


初步技术数据 ADSP-21535

概要

300 MHz 高性能 Blackfin DSP 内核
2 个 16 位 MAC，2 个 40 位 ALU，2 个 40 位累加器，
4 个 8 位视频 ALU，以及 1 个 40 位移位器
RISC 式寄存器和指令模型，编程简单，编译环境友好
先进的调试、跟踪和性能监视
内核电压 V_{DD} 范围 0.9-1.5V，带有动态功率管理
3.3V I/O
商业外壳温度范围 0°C-85°C
工业外壳温度范围（200MHz）-40°C-105°C
260 引脚 PBGA 封装

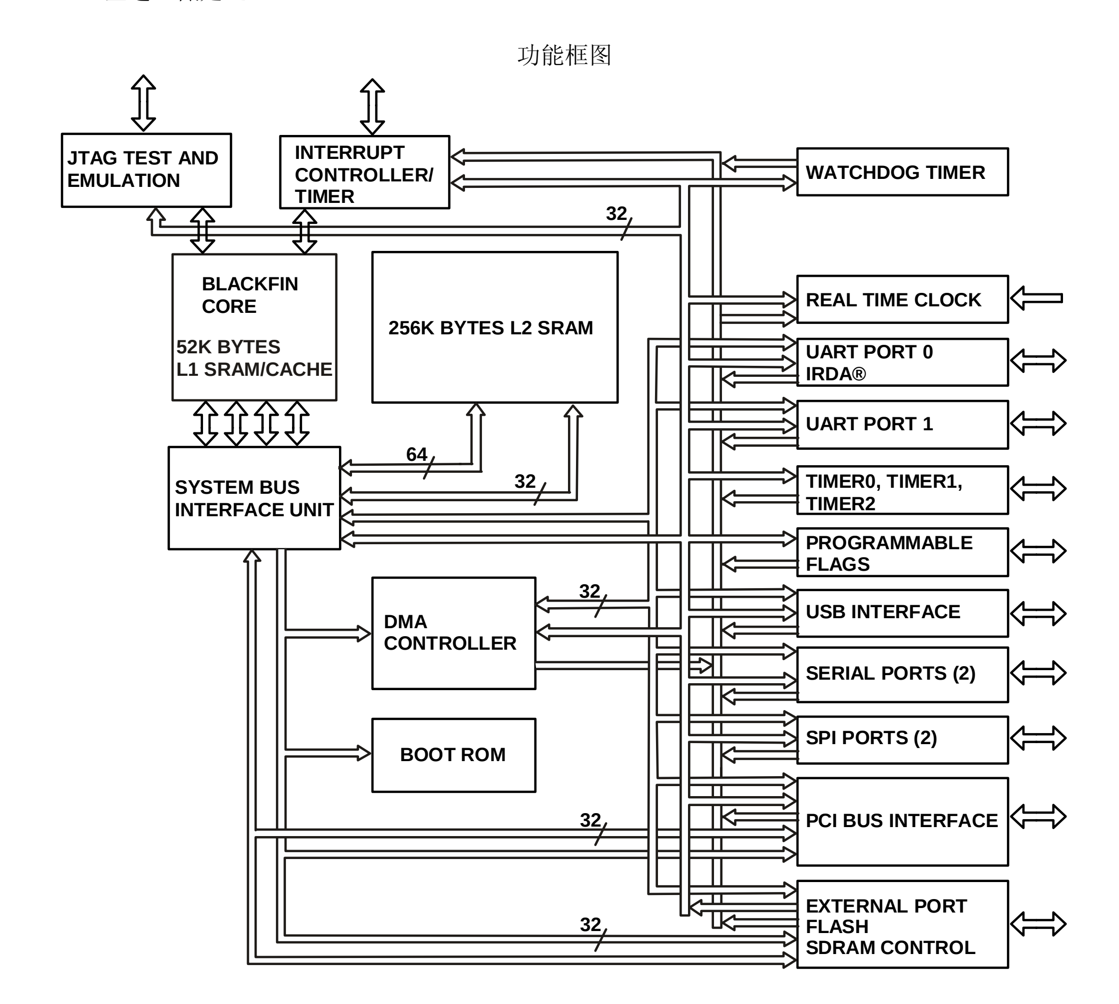
存储器

4GB 统一寻址范围
308KB 片内存储器：
16KB 指令 SRAM/Cache
32KB 数据 SRAM/Cache
4KB 存放中间结果的 SRAM
256KB 全速、低延迟 SRAM
存储器 DMA 控制器
存储器管理单元提供存储器保护
无硬件要求的外部存储控制器
支持同步 SDRAM
支持异步 SRAM、Flash、ROM

外设

32 位、33MHz、3.3V、版本 2.2 兼容的主/从 PCI 总线接口
集成的 USB 1.1 兼容的设备接口
2 个 UART，其中一个支持 IrDA
2 个 SPI 兼容接口
2 个全双工同步串行接口(SPORT)
4 个定时/计数器，其中 3 个支持 PWM
16 个双向可编程标志 I/O 引脚
“看门狗”定时器
实时时钟
1x-31x 倍频的片内 PLL

功能框图



ADSP-21535

2002 年 6 月

说明

本数据手册提供的是关于 ADSP-21535 Blackfin DSP 的初步信息。

概述

ADSP-21535 是 Blackfin DSP 系列产品的成员之一，由 AD 公司和 Intel 公司共同开发，采用了 Micro Signal Architecture (MSA)。这种体系结构将艺术级的 dual-MAC DSP 引擎，简洁的 RISC 式微处理器指令集的优点，以及单指令多数据(SIMD)多媒体能力结合起来，形成了一套独特的指令集结构。

通过集成业界领先和丰富的系统外设及存储器，Blackfin DSP 系列成为下一代需要将 RISC 式编程、多媒体支持和前沿的信号处理等集成在片内的 DSP 的选择平台。

便携式低功耗结构

同其它数字信号处理器相比，Blackfin DSP 系列具有世界领先的低功耗和性能。Blackfin DSP 采用低功耗和低电压的设计方法，具有动态功率管理的特点，即通过改变工作电压和频率来降低总功耗。与仅改变工作频率相比，既改变电压又改变频率能够使总功耗减少 1/3。对于便携式应用来说，这相当于延长了电池的寿命。

系统集成

对下一代的数字通信和便携式的 Internat 应用来说，ADSP-21535 是一个高度集成的片上系统解决方案。通过将工业标准接口与高性能的数字信号处理内核相结合，用户可以快速开发出节省成本的解决方案，而无需昂贵的外部组件。ADSP-21535 系统外设包括 UARTs、SPIs、SPORTs、定时器、实时时钟、可编程标志、看门狗定时器，以及用于无缝外设扩展的 USB 和 PCI 总线。

ADSP-21535 外设

ADSP-21535 包含丰富的外设，它们通过不同的高速宽带内总线与内核相连，使系统不但配置灵活而且有极好的性能（见第 1 页的功能框图）。基本的外设用于一些通用的功能，如 UARTs、带有 PWM（脉冲宽度调制器）和脉冲测量能力的定时器、通用的 I/O 标志引脚、一个实时时钟和一个看门狗定时器。这些外设满足了典型系统的各种需求，并且通过它们增强了系统的扩充能力。除了这些通用的外设，ADSP-21535 还包含有用于各种音频和 modem CODEC 功能的高速串行端口，用于灵活地管理来自片内外设和外部信源的中断事件处理器，以及可根据不同的应用来配置系统的性能和功耗的功率管理控制功能。

在许多系统设计中，用户可以用很少或根本不用其它逻辑电路就可以扩充片内的外设，这是由于 ADSP-21535 提供给用户的一些外设接口符合工业标准总线。这些接口包括一个 32 位、33MHz、V2.2 兼容的 PCI 总线、SPI 串行扩展端口和一个 USB 设备端口。这使得在不同的应用中为连接各种各样的外设而进行系统设计的复杂性已经降到最小了。

除可编程标志引脚、实时时钟和定时器外，所有其它的外设都有一个灵活的 DMA 结构和一个独立的 DMA 通道。片内还有一个独立的存储器 DMA 通道，专用于在 DSP 的不同存储空间（包括外部的 SDRAM 和异步存储器）、内部的 Level 1 和 Level 2 SRAM 和 PCI 存储空间之间的数据传输。片内的多条 32 位总线能以 133MHz 的速度运行，这样提供了足够的带宽以保证处理器内核能够跟得上片内的和片外的外设。

Blackfin DSP 内核

如图 1 所示，Blackfin DSP 内核包含 2 个乘法器/累加器(MAC)，2 个 40 位的 ALU，4 个视频 ALU 和 1 个移位器。ALU 处理来自寄存器组的 8 位、16 位或者 32 位数据。每个时钟周期，每个 MAC 都完成一次 16 位乘 16 位的乘法，并同累加器相加得到一个 40 位宽度、8 位扩展精度的结果。

ALU 执行一套标准的算术和逻辑运算。片内有 2 个 ALU，因此可完成 16 或 32 位数据的运算，这样的灵活性可以满足信号处理各种应用的需求。2 个 32 位输入寄存器中的每一个都可以看作 2 个 16 位的寄存器，因此，每一个 ALU 都能够非常灵活地完成单独的 16 位算术运算。把寄存器看作一对 16 位操作数，在一个指令周期可以完成双 16 位或单 32 位运算。进一步使用第二个 ALU，可以方便地完成 4 个 16 位运算，从而加速每个周期的吞吐量。

功能强大的扩展 40 位移位器可以完成数据的移位、循环移位、标准化、提取和存储等操作。

运算单元的数据来自于多端口的 16 乘 16 位或 8 乘 32 位的寄存器组。

功能强大的程序控制器控制着指令执行的顺序，包括取指和译码。程序控制器支持条件跳转、子程序调用和零耗循环。一个循环缓冲器存储当前的指令，节省了指令存储器的访问时间。

2 套数据地址产生器(DAG)为同时从存储器中存取双操作数提供地址。它们共享一个寄存器组，这个寄存器组由 4 套 32 位的索引、修改、长度和基址寄存器组成。另外的 8 个 32 位寄存器提供用于指向变量和堆栈位置的指针。

Blackfin DSP 采用改进的哈佛结构和分级的存储器结构。Level1(L1)存储器一般以全速运行，没有或只有很少的延迟。Level2(L2)存储器同其它的片内或片外存储器一样，需要花费多个处理器周期来访问。在 Ll 级，指令存储器只存放指令。2 个数据存储器用于存放数据，专用的临时数据存储器存放堆栈和局部变量。L2 级存储器采用统一的存储空间，可以存放指令和数据。

此外，L1 指令存储器和 L1 数据存储器可以配置成静态 RAM(SRAM)或者 CACHE。存储器管理单元(MMU)提供存储器保护功能，对运行于内核上的独立的任务，可保护系统寄存器免于意外的存取。

这种体系结构提供了 3 种运行模式：用户模式、管理员模式和仿真模式。用户模式限制对某些系统资源的访问，因此提供了一个受保护的软件环境；而管理员模式对系统和内核资源的访问不受限制。

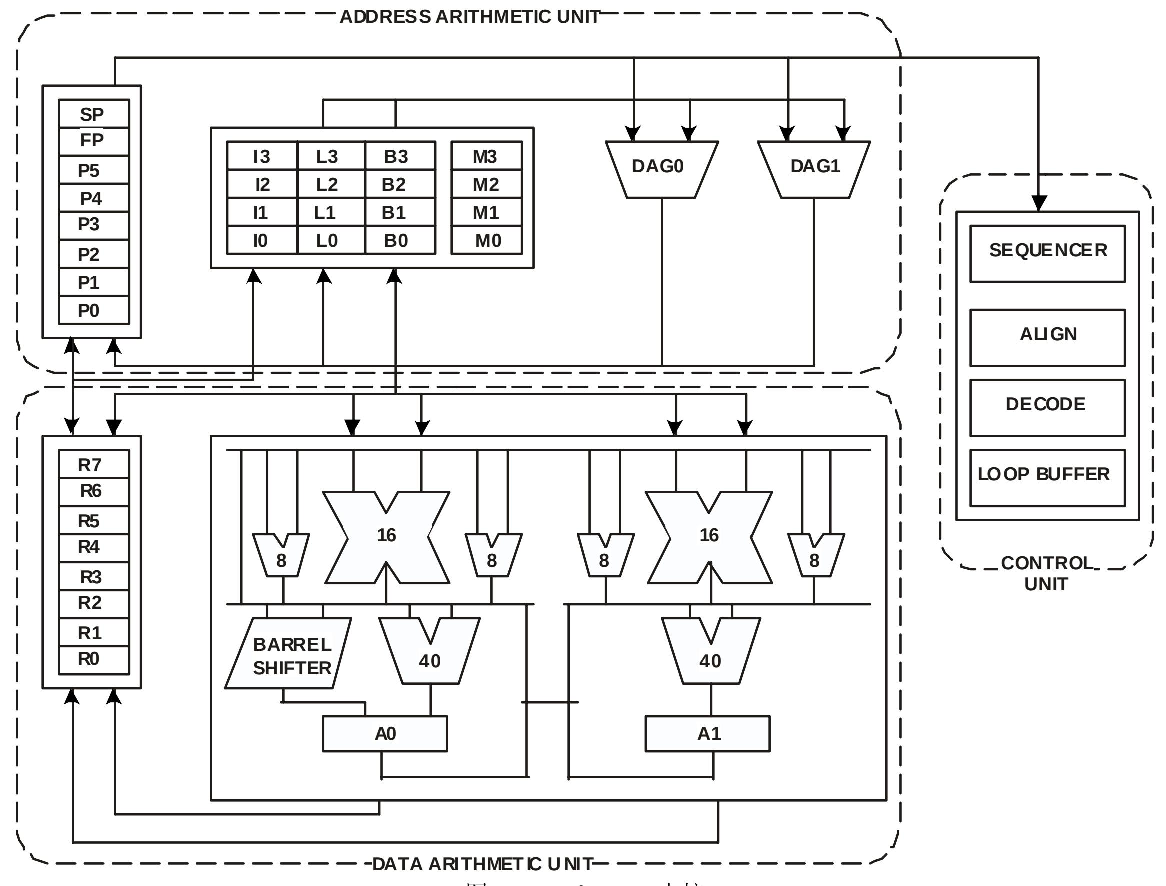


图 1. Blackfin DSP 内核

Blackfin DSP 指令系统经过优化，16 位操作码组成了最常用的指令，这使得编译后的代码密度非常高。复杂 DSP 指令采用 32 位操作码，体现了多功能指令的全部特征。Blackfin DSP 支持有限的并发能力，即 1 个 32 位的指令可以和 2 个 16 位指令并发执行，使编程人员在单指令周期中使用尽可能多的内核资源。

Blackfin DSP 汇编语言使用易于编程和可读性强的代数语法，而且在和 C 编译器的链接上进行了优化，给程序员提供了快速有效的软件环境。

存储器结构

ADSP-21535 把存储器视为一个统一的 4GB 的地址空间，使用 32 位地址。所有的资源，包括内部存储器、外部存储器、PCI 地址空间和 I/O 控制寄存器，都占据公共地址空间的各自独立的部分。此地址空间的各部分存储器按分级结构排列，以提供高的性能价格比。一些非常快速、低延迟的存储器（如 CACHE 或 SRAM）的位置非常接近处理器，而更大的低成本、低性能的存储器远离处理器。参见图 2。

L1 存储器是 Blackfin DSP 内核中性能最高的最重要的存储器。L2 存储器性能稍低，提供额外的空间。最后，通过外部总线接口单元(EBIU)，片外存储器可以由 SDRAM、FLASH 和 SRAM 进行扩展，可以访问超过 768MB 的物理存储器。

存储器的 DMA 控制器提供高带宽的数据传输能力。它能够在内部的 L1/L2 存储器和外部存储器空间（包括 PCI 存储空间）之间执行代码或数据的块传输。

内部（片内）存储器

ADSP-21535 有 4 块片内存储器，提供到内核的高带宽的访问。

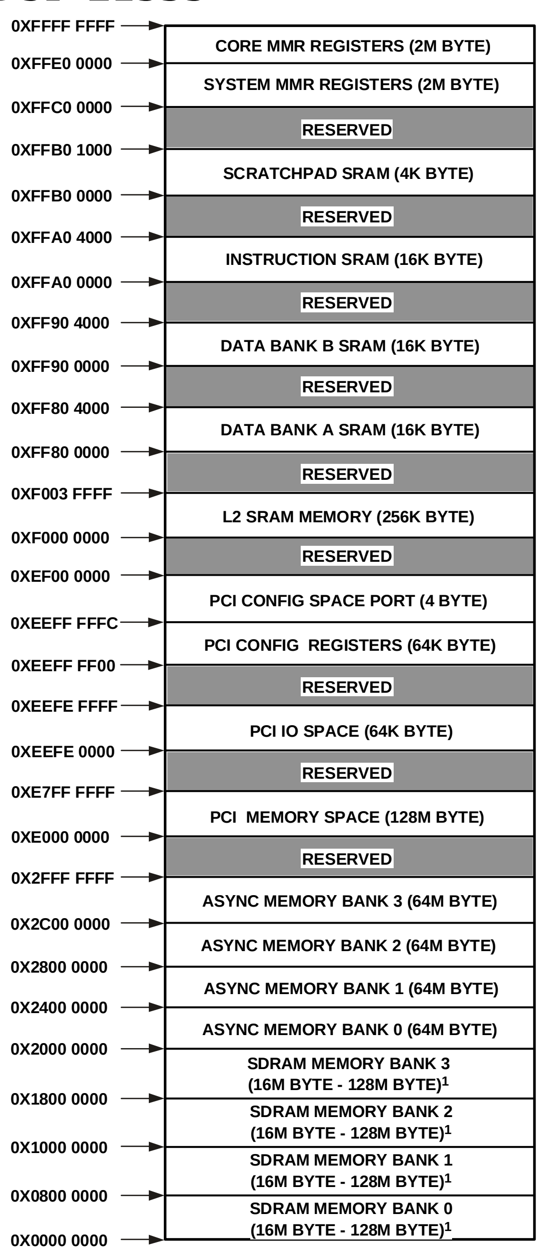
第 1 块是 16KB 的 L1 指令存储器，可配置成 4 路组联合的 CACHE，也可配置成 SRAM，它以处理器的最快速度访问。

第 2 块片内存储器是由 2 块 16KB 组成的 L1 数据存储器。每个 L1 数据存储器都能配置成单路或双路组联合的 CACHE 或者 SRAM，同样被内核以最快速度访问。

第 3 块存储器模块是一个 4KB 的临时数据 RAM，它和 L1 存储器有相同的运行速度，但是只能作为数据 SRAM（不能配置为 CACHE，也不能通过 DMA 访问）。

第 4 块片内存储器是 256KB 的高速 L2 SRAM，同样以内核全带宽运行，和 L1 存储器相比，延迟稍长。L2 存储器是一个统一的指令和数据存储器，能够根据系统设计要求同时存放代码和数据。

Blackfin DSP 内核有一个专用的低延迟 64 位宽的数据通道进入 L2 SRAM 存储器。例如，在内核 300MHz 频率下，通过此接口的峰值数据传输率达到每秒 2.4GB。



[1]THE ADDRESSES SHOWN FOR THE SDRAM BANKS REFLECT A FULLY POPULATED SDRAM ARRAY WITH 512M BYTES OF MEMORY. IF ANY BANK CONTAINS LESS THAN 128M BYTES OF MEMORY, THAT BANK WOULD EXTEND ONLY TO THE LENGTH OF THE REAL MEMORY SYSTEMS, AND THE END ADDRESS WOULD BECOME THE START ADDRESS OF THE NEXT BANK. THIS WOULD CONTINUE FOR ALL FOUR BANKS, WITH ANY REMAINING SPACE BETWEEN THE END OF MEMORY BANK 3 AND THE BEGINNING OF ASYNC MEMORY BANK 0, AT ADDRESS 0X2000 0000, TREATED AS RESERVED ADDRESS SPACE.

图 2. 内部/外部存储器映射

外部（片外）存储器

外部存储器通过外部总线接口单元(EBIU)进行访问。此接口可与最多 4 个同步 DRAM（SDRAM），或与最多 4 个异步存储器设备（包括 FLASH、EPROM、ROM、SRAM 和存储器映射 I/O 设备）直接连接。

PC133 兼容的 SDRAM 控制器可以被编程并有与最多 4 个 bank 的 SDRAM 接口，每个 bank 的容量在 16MB 到 128MB 之间，提供给系统最多达 512MB 的 SDRAM 空间。每个 bank 空间都是可独立编程的，无论大小及其位置是否相同，相邻的 bank 地址都是连续的。这样就允许灵活地配置和升级系统的存储器，并且允许内核把所有的 SDRAM 视为一个简单、连续的物理地址空间。

异步存储器的控制器也能够通过编程控制多达 4 个 bank 的时序参数灵活的各种异步存储设备。无论使用设备的大小如何，每个 bank 的空间都占据 64MB。这样，只有装满 4 个 64MB 的存储器时地址空间才能连续。

PCI

PCI 总线定义了 3 个独立的地址空间，这些地址空间通过 ADSP-21535 存储空间的窗口进行访问。它们分别是 PCI 存储器、PCI I/O 和 PCI 配置空间。

此外，PCI 接口可以有两种配置方法。其一，本处理器用作系统中的控制 CPU，PCI 接口用作与它相接的 PCI 桥；其二，系统中另有一个 CPU 作为主机，本 ADSP-21535 用作 PCI 总线上的智能 I/O 设备。

当 ADSP-21535 作为系统控制器时，它通过映射窗口来查看 PCI 地址空间，并且能初始化系统中的所有设备，维持环境的拓扑映射。

PCI 存储区域是出现在 PCI 总线上的 4GB 的存储空间，可用来映射总线上的存储器 I/O 设备。ADSP-21535 使用存储空间中的一个 128MB 窗口来查看 PCI 存储空间。同时使用一个基地址寄存器，在 4GB PCI 存储空间中定位窗口的起始位置，而在处理器地址空间中这个窗口的地址保持不变。

PCI I/O 区域也是 4GB 空间。然而，对于 I/O 映射地址，大多数系统和 I/O 设备只使用这个空间的一个 64KB 子集。ADSP-21535 用一个基地址寄存器把一个 64KB 的窗口实现于这个空间，虽然可以定位于 PCI I/O 地址空间的任何地方，但是这个窗口在处理器地址空间处于固定位置。

PCI 配置空间是一个有限的地址空间，用于系统列举和初始化，在处理器与 PCI 设备之间这是一种低性能的通信模式。ADSP-21535 提供一个单值窗口，在 PCI 配置空间中的任何地址都可存取一个数据的值。这个窗口是固定的，写操作时接收这个数据的地址和的值，而读操作时把这个值返回到同一地址。

I/O 存储器空间

Blackfin DSP 没有定义独立的 I/O 空间。所有的资源都被映射到统一的 32 位地址空间。片上 I/O 设备的控制寄存器被映射到靠近 4GB 地址空间顶端的存储器映射寄存器(MMR)地址范围内。这个地址空间又被划分为 2 个部分，一部分包含完成所有内核功能的控制 MMR，另一部分包含用于设置和控制内核以外的片内外设的寄存器。内核 MMR 仅在管理员模式下可被内核访问，对于片内外设和通过 PCI 总线访问的外部设备，内核 MMR 都被保留起来。系统 MMR 在管理员模式下可被内核访问，而且根据所需的系统保护模式，系统 MMR 可对其它设备可见或保留。

引导

ADSP-21535 包括一个小的引导内核，用于配置的适当的外设。如果 ADSP-21535 被配置为从引导 ROM 存储

器引导，那么 DSP 从片内引导 ROM 开始执行。若需要更多信息，请看第 12 页的引导模式。

事件处理

ADSP-21535 的事件控制器处理到达处理器的所有的同步和异步事件。ADSP-21535 事件处理支持嵌套和优先级。嵌套允许同时激活多个事件的服务程序。优先级保证高优先级事件的响应可以抢占较低优先级事件的响应。控制器支持 5 种类型的事件：

1. 仿真：仿真事件使处理器进入仿真模式，允许通过 JTAG 接口命令和控制处理器。
2. 复位：此事件使处理器复位。
3. 不可屏蔽中断(NMI)：NMI 事件可以由软件"看门狗"定时器或者处理器的 NMI 输入信号产生的。NMI 事件经常用作断电指示，有序地进行系统关闭工作。
4. 异常：异常是与程序执行同步发生的事件，即指令执行完之前可能会产生异常。例如数据对准违规、未定义指令等情况都将导致异常。
5. 中断：中断是与程序执行异步发生的事件，由定时器、外设、输入引脚等引起。

每个事件都有一个相应的保存返回地址的寄存器和一个相应的从事件返回指令。一个事件被触发后，处理器当前状态被保存在管理员堆栈内。

ADSP-21535 事件控制器包括 2 个部分，内核事件控制器(CEC)和系统中断控制器(SIC)。内核事件控制器和系统中断控制器协同工作来确定优先级和控制所有系统事件。从概念上讲，来自外设的中断进入到 SIC，然后被直接发送到 CEC 的通用中断中处理。

内核事件控制器(CEC)

除专用中断和异常事件外，CEC 还支持 9 个通用中断(IVG15-7)。这些通用中断中，推荐将优先级最低的 2 个中断(IVG15-14)留作软件中断，剩下的 7 个优先级中断分别用于 ADSP-21535 的外设。表 1 描述了 CEC 的事件、事件向量表(EVT)的名称及优先权。

系统中断控制器(SIC)

系统中断控制器为来自不同外设的中断源提供至 CEC 通用中断输入的映射和路由。尽管 ADSP-21535 提供了默认的映射，用户仍可以通过改写中断设置寄存器(IAR)的值，来改变中断事件的映射和优先权。表 2 描述了 SIC 的中断源和至 CEC 的默认映射。

事件控制

ADSP-21535 为用户提供了非常灵活的机制来控制事件的处理。在 CEC 中，有 3 个寄存器用于调整和控制事件。如下所示，它们中的每个寄存器都是 16 位宽度，每一位都代表一种特殊的事件分类：

表 1. 内核事件控制器(CEC)

优先权（0 为最高）	事件类	EVT 入口
0	仿真/测试	EMU
1	复位	RST
2	不可屏蔽	NMI
3	异常	EVX
4	全局使能	-
5	硬件错误	IVHW
6	内核定时器	IVTMR
7	通用中断 7	IVG7
8	通用中断 8	IVG8
9	通用中断 9	IVG9
10	通用中断 10	IVG10
11	通用中断 11	IVG11
12	通用中断 12	IVG12
13	通用中断 13	IVG13
14	通用中断 14	IVG14
15	通用中断 15	IVG15

表 2. 系统中断控制器(SIC)

外设中断事件	外设中断 ID	默认映射
实时时钟	0	IVG7
保留	1	-
USB	2	IVG7
PCI 中断	3	IVG7
SPORT 0 Rx DMA	4	IVG8
SPORT 0 Tx DMA	5	IVG8
SPORT 1 Rx DMA	6	IVG8
SPORT 1 Tx DMA	7	IVG8
SPI 0 DMA	8	IVG9
SPI 1 DMA	9	IVG9
UART 0 Rx	10	IVG10
UART 0 Tx	11	IVG10
UART 1 Rx	12	IVG10
UART 1 Tx	13	IVG10
定时器 0	14	IVG11
定时器 1	15	IVG11
定时器 2	16	IVG11
GPIO 中断 A	17	IVG12
GPIO 中断 B	18	IVG12
存储器 DMA	19	IVG13
软件看门狗定时器	20	IVG13
保留	26-21	-
软件中断 1	27	IVG14
软件中断 2	28	IVG15

1. CEC 中断锁存寄存器(ILAT)：ILAT 寄存器用于指示事件已被锁存。处理器锁存事件后相应的位置 1，事件被

系统接受后该位清 0。该寄存器被控制器自动刷新，但仅在管理员模式下可读。

2. CEC 中断屏蔽寄存器(IMASK)：IMASK 寄存器控制发生的事件是否被屏蔽。当 IMASK 寄存器的相应位置 1 时，事件不被屏蔽，发生后由系统处理。该位清 0 将屏蔽事件，即使该事件已被锁存在 ILAT 寄存器中，处理器也不会处理该事件。在管理员模式下，该寄存器可以被读写。（注：通用中断可以通过 STI 和 CLI 指令设置为全局使能和禁止。）

3. CEC 中断等待寄存器(IPEND)：IPEND 寄存器跟踪所有嵌套的事件。IPEND 寄存器中的相应位置 1 表示事件当前处于活动状态或嵌套在某一级。该寄存器被控制器自动刷新，但是在管理员模式下才能读取。

SIC 使用 3 个 32 位中断控制和状态寄存器，来进一步控制事件的处理。每个寄存器都包含与表 2 所示的每个外设中断相对应的位。

1. SIC 中断屏蔽寄存器(SIC_IMASK)：此寄存器控制每个外设中断事件是否被屏蔽。当寄存器的相应位置 1 时，事件不被屏蔽，发生后由系统处理。该位清 0 将屏蔽外设事件，使之不被处理。

2. SIC 中断状态寄存器(SIC_ISTAT)：由于多个外设可以映射到同一事件，该寄存器允许软件设置哪个外设事件源触发该中断。相应位置 1 表明外设发出了中断，为 0 则表明外设未发出事件。

3. SIC 中断唤醒使能寄存器(SIC_IWR)：通过使能该寄存器中的相应位，当事件发生而处理器处于掉电模式时，每个外设都可唤醒处理器。（更多信息，请见第 9 页动态功率管理。）

由于多个中断源可以映射到同一个通用中断，因此该中断输入引脚上可能同时出现多个脉冲，这可以发生在对一个已检测到的中断处理之前或之中。IPEND 寄存器的内容由 SIC 监控，以检查中断是否得到确认。

当一个中断上升沿被检测到（检测需要 2 个处理器时钟周期），ILAT 寄存器的相应位被置 1。当 IPEND 寄存器的任一位被置 1 时，该位被清 0。IPEND 的这位表示该事件已进入处理器流水线。此时，CEC 将在下一个事件到来时识别其上升沿，并将这一事件排入队列。从通用中断的上升沿到 IPEND 寄存器的输出置 1，最小的延迟为 3 个处理器时钟周期；然而，由于内部的活动和处理器的模式不同，延迟可能更长。

DMA 控制器

ADSP-21535 有多个独立的 DMA 控制器，能够以最小的 DSP 内核开销完成自动的数据传输。DMA 传输可以发生在 ADSP-21535 的内部存储器和任一有 DMA 能力的外设之间。此外，DMA 传输也可以在任一有 DMA 能力的外设和已连接到外部存储器接口（包括 SDRAM 控制器、异步存储器控制器和 PCI 总线接口）的外部设备之间完成。有 DMA 传输能力的外设包括 SPORT、SPI 端口、UART 和 USB 端口。每个独立的有 DMA 能力的外设至少有一个专用 DMA 通道。PCI 的 DMA 传输由存储器的 DMA 通道来完成。

为了描述每个 DMA 序列，DMA 控制器使用一套名为描述子块的参数。当需要后继的 DMA 序列时，这些描述子块被链接或串接起来，这样，一个 DMA 序列的完成能够自动初始化并启动下一个序列。为能访问整个 ADSP-21535 的地址空间，原地址和目标地址描述子块采用了全 32 位地址的基指针。

除专用外设的 DMA 通道以外，在 ADSP-21535 的不同存储器之间有一个独立的存储器 DMA 通道。这使得任意的存储器（包括片内 L2 存储器、外部 SDRAM、ROM、SRAM 和 FLASH）与 PCI 地址空间之间的数据块传输成为可能，几乎不需处理器干预。

外部存储器控制

ADSP-21535 的外部总线接口单元(EBIU)，为各种工业标准的存储器设备提供一个高性能、无需附加硬件的接口。该控制器由两部分组成：第一部分是用于连接工业标准的同步 DRAM 设备和 DIMM 的 SDRAM 控制器，第二部分是用于与各种存储设备接口的异步存储控制器。

PC133 SDRAM 控制器

SDRAM 控制器能以 f_{SCLK} 的速度，与多达 4 个 bank 的工业标准 SDRAM 或 DIMM 接口。每个 bank 可以配置为 16MB 至 128MB 的存储器，完全符合 PC133 SDRAM 标准。

即使不同的 bank 使用不同大小的存储器，控制器也能把所有 bank 管理成一个连续的地址空间，以便处理器将它视为单一地址空间。这使初始配置的系统设计在将来便于用类似的或不同的存储器进行升级。

一套可编程的时序参数用于配置 SDRAM bank，可以支持更低速的存储设备。存储器 bank 可以配置成最高性能和带宽的 32 位宽度或者低成本的 16 位宽度设备。

所有 4 个 bank 使用相同的 SDRAM 控制信号和独立的 bank 选通信号，为大多数的系统配置提供了一个完整的无缝接口。

异步控制器

异步存储器控制器提供与多达 4 个 bank 的存储器或 I/O 设备的可配置的接口。每个 bank 都可以用不同的时序参数独立编程，使之与各种存储设备（包括 SRAM、ROM 和 FLASH EPROM）及与标准存储器控制线接口的 I/O 设备相连接。每个 bank 在处理器地址空间中都占据一个 64MB 的窗口，但如果每个 bank 容量不满 64MB 就不能组成一个连续的地址空间。这些 bank 也可以配置为低成本和功耗的 16 位总线宽度或高性能的 32 位总线宽度，以简化与各种存储器和 I/O 设备的接口。

PCI 接口

ADSP-21535 提供一个 33MHz，3.3V，32 位，PCI 版本 2.2 兼容的无需其它硬件的逻辑和硬件接口。该 PCI 接

口设计为 3 伏的信号环境，在存储器内核和片上外设及外部 PCI 总线之间提供了一个总线桥接功能。ADSP-21535 的 PCI 接口支持如下 2 种 PCI 功能：

1. 主机到PCI桥功能，从PCI目标设备来看，ADSP-21535 的资源（处理器内核、片内和片外存储器、存储器 DMA 控制器）提供了必要的硬件部件，来仿真 PC 主机的 PCI 接口。
2. PCI 目标功能，基于 ADSP-21535 的智能外设可以容易地与版本 2.2 兼容的 PCI 总线接口。

PCI 主机功能

用作 PCI 主机时，ADSP-21535 提供必要的 PCI 主机（平台）功能，以支持和控制各种流行的 PCI I/O 设备（如以太网控制器、总线桥等）。

注意，Blackfin DSP 体系结构中只定义了存储器空间（没有 I/O 或配置地址空间）。PCI 的 3 个地址空间（存储器、I/O 和配置空间）都被映射到 ADSP-21535 的统一的 32 位存储空间中。由于 PCI 存储器空间与 ADSP-21535 存储器地址空间大小相同，所以使用一个加窗的方法，在 ADSP-21535 地址空间中采用独立的窗口用来访问 3 个 PCI 地址空间。基地址寄存器用于定位这些窗口，以便能够访问 PCI 地址空间中的任何区域，而它们在 ADSP-21535 处理器地址空间内却保持固定位置。

对于欲查看 ADSP-21535 资源的 PCI 总线设备，提供了几个映射寄存器，以使在 PCI 地址空间中查看这些资源成为可能。ADSP-21535 的外部存储器空间、片内 L2 和某些 I/O MMR 可以被选择使能，用作 PCI 总线设备的目标地址空间，用于 PCI 存储器操作。

PCI 目标功能

用作为一个 PCI 目标设备时，PCI 主机处理器在列举 PCI 总线系统时，可配置 ADSP-21535 子系统。一经配置，该子系统就成为一个智能的 I/O 设备。被配置为目标设备后，PCI 控制器使用存储器 DMA 控制器来完成 PCI 主机要求的 DMA 传输。

USB 设备

ADSP-21535 提供一个 USB1.1 兼容的设备类型接口，支持直接与主机系统相连接。USB 内核接口提供一个灵活的多达 8 个终端的可编程环境。每个终端都能支持所有 USB 数据传输类型（包括控制、批量、中断和等时）。每个终端都为传输数据提供了存储器映射的缓冲器。ADSP-21535 USB 端口有一个专用的用于最小化处理器轮询开销的 DMA 控制器和仅当传输完成时才向 CPU 发出异步请求的中断输入。

实时时钟

ADSP-21535 的实时时钟(RTC)提供了一个具有当前时间、跑表和报警等功能的稳定的数字表。该 RTC 的时钟采用 ADSP-21535 外部的 32.768KHz 晶振，RTC 有专用的电源引脚，以使得当处理器其它部分处于低功耗状态时 RTC 仍然保持供电和时钟。RTC 提供了数个可编程的中断选择（包括以时、分、秒计数中断，可编程跑表倒数计数中断，或者已编程的警报时钟中断）。

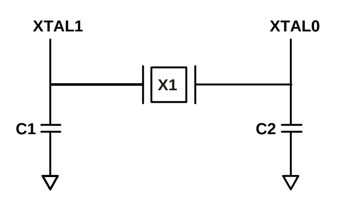
32.768KHz 的输入时钟频率通过分频器成为 1Hz 信号。具有计数功能的定时器包括 4 个计数器：一个 6 位的秒计数器、一个 6 位的分计数器、一个 5 位的小时计数器和一个 8 位的日计数器。

报警功能启动后，当定时器的输出和报警控制寄存器中给定值相等时，报警功能会产生一个中断。报警分为 2 类：第一类是时间报警；第二类是日期加时间报警。

跑表功能从一个已编程的值起以 1 分钟分辨率倒数计数。当跑表功能启动且计数器下溢出时，便产生中断。

同其它外设一样，RTC 能在产生任何中断之前把 ADSP-21535 处理器从低功耗状态唤醒。

用外部元件连接 RTC 的引脚 XTALI 和 XTALO，如图 3 所示。



SUGGESTED COMPONENTS:
ECLIPTEK EC38J (THROUGH-HOLE PACKAGE)
EPSON MC405 12 pF LOAD (SURFACE MOUNT PACKAGE)
C1 = 22 pF
C2 = 22 pF

NOTE: C1 AND C2 ARE SPECIFIC TO CRYSTAL SPECIFIED FOR X1.
CONTACT CRYSTAL MANUFACTURER FOR DETAILS. C1 AND C2 SPECIFICATIONS ASSUME BOARD TRACE CAPACITANCE OF 3 pF.

推荐元件：
ECLIPTEK EC38J（通孔封装）
EPSON MC-405 12.5pF LOAD(表贴封装)
C1=22pF
C2=22pF
注意：C1 与 C2 专用于推荐的晶振 X1
联系晶振制造商索取 C1 和 C2 的详细资料
规格假定板上线路容值为 3pF

图 3. RTC 的外部元件

“看门狗”定时器

ADSP-21535 包含一个 32 位定时器，可用于执行软件的“看门狗”功能。软件“看门狗”可以提高系统的可靠性，如果在软件复位前定时器溢出，软件“看门狗”通过产生一个硬件复位、不可屏蔽中断(NMI)或通用中断来强迫处理器进入一个已知状态。程序员初始化定时器计数值，使能相应的中断，然后启动定时器。随后，软件必须在计数器从给定值计数到 0 前重新装载计数器。这样防止系统停留在未知状态。在未知状态下，软件由于外部噪声或者软件错误等停止运行后，通常将定时器复位。

复位后，软件可以通过查询定时控制寄存器的一个状态位来确定“看门狗”是否为硬件复位的来源，该位仅当

“看门狗”产生复位时置 1。

定时器的时钟采用系统时钟(SCLK)，以最高频率 f_{SCLK} 运行。

定时器

ADSP-21535 有 4 个可编程定时器。3 个通用定时器都连有外部引脚，可以用作脉冲宽度调制器（PWM）或定时器输出，也可以用作定时器的输入时钟或测量外部事件的脉冲宽度的输入。3 个通用定时器中的每一个都能够独立地编程用作 PWM、内部或外部时钟驱动的定时或者脉冲宽度计数器。

通用定时器单元可以与 UART 联合使用，进行串行通道数据流的脉冲宽度的测量，提供自动的波特率检测功能。

通用定时器能够向处理器内核发出中断，为同步、处理器时钟、或外部信号的计数值提供周期性事件。

除 3 个通用可编程定时器外，ADSP-21535 还提供了第 4 个定时器。这个额外的定时器由内部处理器时钟(CCLK)驱动，一般用作系统标记时钟，用以产生操作系统的周期性中断。

串行口(SPORT)

ADSP-21535 提供 2 个完整的同步串行端口（SPORT0 和 SPORT1）来完成串行和多处理器的通信工作。SPORT 有以下特点：

1. 双向操作：每个 SPORT 都有独立的发送和接收引脚。
2. 缓冲的（8-deep）发送和接收端口：每个端口都有一个数据寄存器，用以同其它 DSP 部件间进行双向数据传输；多个移位寄存器用于数据寄存器内数据的移位。
3. 时钟：每个发送或接收端口或者使用外部串行时钟，或者使用自己产生的频率范围为 (f_{SCLK} /131070) Hz 到(f_{SCLK} /2) Hz 的时钟。
4. 字长：每个 SPORT 都支持 3 到 16 位长度的串行数据字，以最高有效位在前或最低有效位在前的格式传送。
5. 帧：每个发送和接收端口在无论每个数据字有无帧同步信号的情况下都能运行。帧同步信号能够从内部或者外部产生，可以高有效或低有效，要求 2 个脉冲宽度，可以前或后帧同步。
6. 硬件压扩：每个 SPORT 都能根据 ITU 推荐的 G.711 标准完成 A 律和 μ 律压扩。对 SPORT 的发送和/或接收通道都能进行压扩，而且没有额外的延迟。
7. 单时钟周期开销的 DMA 操作：每个 SPORT 都可以自动发送和接收多个缓冲区的数据信息。DSP 能够链接或串接 SPORT 和存储器之间的多个 DMA 序列。串接后的 DMA，通过建立串接的描述子块，能够进行动态的分配和更新。
8. 中断：完成数据传输或者通过 DMA 传输完整个数据缓冲区之后，每个发送和接收端口都能产生一个中断。
9. 多通道能力：每个 SPORT 支持 128 通道，并兼容 H.100、H.110、MVIP-90 和 HMVIP 标准。

串行外设接口(SPI)

ADSP-21535 有 2 个 SPI 兼容的端口，能够与多个 SPI 兼容的设备通信。

SPI 接口使用 3 个引脚传输数据：2 个数据引脚（主输出-从输入 MOSIx 和主输入-从输出 MISOx）和 1 个时钟引脚（串行时钟 SCKx）；2 个 SPI 片选输入引脚（$\overline{\text{SPISSx}}$）可使其它 SPI 设备选择 DSP；14 个 SPI 片选输出引脚（SPIxSEL7-1）使 DSP 能够选择其它 SPI 设备。这些 SPI 引脚也可以被重新配置为可编程标志引脚。通过这些引脚，SPI 端口提供了全双工的同步串行接口，支持主从模式和多主环境。

每个 SPI 的波特率和时钟的相位/极性都是可编程的（见图 4），而且每个 SPI 都集成有一个 DMA 控制器，可配置为发送或接收数据流。SPI 的 DMA 控制器在任意给定时间，只能进行单向存取。

$$SPI\ \text{时钟率} = \frac{f_{SCLK}}{2 \times SPIBAUD}$$

图 4. SPI 时钟率计算

在传输过程中，SPI 端口从 2 个串行数据引脚进行串行数据移入和移出，同时完成发送和接收工作。串行时钟引脚使 2 条串行数据引脚上的数据移位和采样同步。

在主模式下，DSP 采用如下的顺序来设置和初始化 SPI 传输：

1. 使能和配置 SPI 端口（数据大小和传输格式）。
2. 用 SPIxSELy 输出引脚（可重新配置为可编程标志引脚），选择从目标 SPI。
3. 在 DSP 存储空间（仅在 DMA 模式下是可选择的）中，定义一个或多个的 TCBs。
4. 使能 SPI DMA 引擎并指定传输方向。
5. 读写 SPI 端口接收或发送的缓冲数据（仅在非 DMA 模式下）。

SCKx 引脚为同时移出 MOSIx 上的数据和移入 MISOx 上的数据产生编程时钟脉冲。仅在 DMA 模式下，SPI DMA 的字计数器从 1 变为 0 时，才停止传输。

在从模式下，DSP 采用如下的顺序来设置 SPI 端口接收主 SPI 设备发送来的数据：

1. 使能和配置 SPI 的从端口，与主 SPI 设备的运行参数（数据大小和传输格式）匹配。
2. 在 DSP 存储空间内定义和产生一个接收 TCB，在数据传输结束时（仅在 DMA 模式下是可选的）时产出中

断。

3. 使能 SPI DMA 以接收数据（仅在 DMA 模式下是可选的）。

4. 当接收到$\overline{\text{SPISSx}}$输入引脚（可重新配置为可编程标志引脚）的 SPI 片选信号后，在适当的 SPI SCKx 边沿，开始接收来自主 SPI 设备的数据。

仅在 DMA 模式下，SPI DMA 的字计数器从 1 变为 0 时，才停止传输。此时若有下一个命令 TCB 在等待，DSP 将继续传输。

从模式的发送操作与此类似，但 DSP 指定的发送数据的数据缓冲区、产生和放弃对发送 TCB 的控制、及开始填充 SPI 端口的数据缓冲区都不相同。如果 SPI 控制器在数据发送时没有准备好，它将发送一个“0”字。

UART 端口

ADSP-21535 提供 2 个全双工的通用异步接收/发送(UART)端口（UART0 和 UART1），与 PC 标准的 UART 完全兼容。UART 端口为其它外设或主机提供了一个简化的 UART 接口，支持全双工、有 DMA 能力的异步串行数据传输。每个 UART 端口支持 5 至 8 个数据位、1 或 2 个停止位和无校验/奇校验/偶校验位的支持。UART 端口支持以下 2 种模式的操作：

1. PIO（编程 I/O）：处理器通过读/写 I/O 映射的 UATX 或 UARX 寄存器，发送或接收数据。在发送和接收时，数据都是双缓冲的。

2. DMA（直接存储器访问）：DMA 控制器传输发送和接收数据。这就减少了与存储器传输数据所需的中断的数量和频率。每个 UART 都有 2 个专用的 DMA 通道，一个用于发送，一个用于接收。这些 DMA 通道的优先权低于大多数 DMA 通道，因为其使用率相对较低。

每个 UART 端口的波特率（见图 5）、串行数据格式、错误代码的产生和状态、及中断均可编程：

1. 支持位速率每秒从（f_{SCLK}/1048576）到（f_{SCLK}/16）位。

2. 支持每帧7至12位的数据格式。

3. 发送和接收操作都可被配置为能产生可屏蔽中断。

$$UART\ \ 时钟频率 = \frac{f_{SCLK}}{16 \times D}$$

图 5. UART 时钟频率计算

这里D=1 to 65536

同通用定时器联合使用可进行自动波特率检测。

UART0的功能被进一步扩展，能支持IrDA串行红外物理层连接规范（SIR）协议

可编程标志（PFX）

ADSP-21535 有 16 个双向的通用可编程 I/O 引脚（PF15-0）。这些引脚还有专用的功能，用于时钟倍频选择、SROM 引导模式和 SPI 端口操作。详见第 8 页的串行外设接口（SPI）和第 11 页的时钟信号。每一个可编程引脚都能通过操作下面的的标志控制寄存器、标志状态寄存器和标志中断寄存器被独立控制：

1. 标志方向控制寄存器—规定每个独立的PFx引脚的方向，用作输入或输出。

2. 标志控制和状态寄存器—ADSP-21535使用一种“写1设置”和“写1清除”的机制控制每个标志的设置，而不是常用的软件读-修改-写操作。这种机制能够在不影响其它标志的情况下，一个处理器周期内完成对这些标志进行任意组合地设置或清除。两个寄存器用于此工作，一个用来设置标志，另一个用来清除标志。通过软件读标志位状态寄存器，可以查看标志的状态。

3. 标志中断屏蔽寄存器—2 个标志中断屏蔽寄存器允许每个独立的 PFx 引脚为处理器提供中断。与用于设置和清除标志的 2 个标志控制寄存器类似，一个标志屏蔽寄存器通过设置相应的位来使能中断功能，另一个标志屏蔽寄存器通过清除相应的位来禁止中断功能。PFx 引脚定义为输入时，能够设置为产生硬件中断；定义为输出时，能够设置为产生软件中断。

4. 标志中断敏感寄存器—2 个标志中断敏感寄存器用于规定每个 PFx 引脚是电平还是边沿敏感，如果是边沿敏感，还规定是仅上升沿敏感，还是信号的上升沿和下降沿都很敏感。一个寄存器选择敏感类型，另一个寄存器选择边沿敏感的敏感边沿。

动态功率管理

ADSP-21535 提供 4 种运行模式，每种模式有不同的性能/功耗特性。动态功率管理用适当的外部电源调节能力来动态地改变处理器内核供电电压，进一步降低功耗。控制每一个 ADSP-21535 外设的时钟也能降低功耗。表 3 概要列出了每种模式下的功率设置。

全速运行模式—最高性能

在全速模式下，PLL 被使能，且不被旁路，因此提供最高运行频率。这是获得最高性能的正常执行状态。处理器内核和所有使能的外设都以全速运行。

活动运行模式—中等功率节省

在此模式下，PLL 被使能，但被旁路。输入时钟(CLKIN)用于为处理器内核(CCLK)和外设(SCLK)产生时钟。由于 PLL 被旁路，CCLK 以 CLKIN 一半的频率运行，此时可节省大量的功率。此模式下，PLL 倍频率可以通过设置 PLL 控制寄存器(PLL_CTL)中的 SSEL 域改变。

在活动运行模式下，系统支持对适当配置的 L1 存储器进行 DMA 访问。

休眠运行模式-高功率节省

休眠运行模式通过关闭处理器内核（CCLK）的时钟脉冲来降低功耗，然而PLL和系统时钟（SCLK）仍在运行。任何中断，一般通过外部事件或RTC活动，将唤醒处理器。此模式下任何中断到来时，处理器都会检查PLL控制寄存器(PLL_CTL)中旁路位（BYPASS）的值。如果旁路位被关闭，处理器将切换到全速运行模式。如果旁路位使能，处理器将切换到活动运行模式。休眠模式时，系统DMA不支持对L1存储器的访问。

深度休眠运行模式—最大功率节省

通过关闭处理器内核（CCLK）和所有同步系统（SCLK）的时钟，深度休眠运行模式将获得最大的功率节省。异步系统，如 RTC，仍将运行但将不能访问内部资源或外部存储器。这种掉电模式只能通过复位中断（$\overline{\text{RESET}}$）或由 RTC 产生的异步中断退出。此模式下$\overline{\text{RESET}}$有效时，处理器会检查 BYPASS 引脚的值。如果旁路被关闭，处理器将切换到全速运行模式。如果旁路使能，处理器将切换到活动运行模式。此模式下 RTC 产生的异步中断有效时，处理器将切换到全速运行模式，而不管 BYPASS 引脚的值。

DEEPSLEEP 引脚的输出在此模式下有效。

模式转换

图 6 给出了可用的模式转换，这些模式转换或者通过本节描述的中断事件完成，或通过对 PLLCTL 寄存器赋值并执行 PLL 指令来完成。

此指令序列在中断禁用时将处理器切换到一种已知的空闲状态。注意在此状态下所有的 DMA 活动都被禁用。

功率节省

如表 4 所示，ADSP-21535 支持 5 种不同的电源范围。使用多个电源范围在与工业标准和惯例兼容的同时，可获得最大的灵活性。通过将 ADSP-21535 内部逻辑隔离为独立的电源，同 PLL、RTC、PCI 和其它 I/O 分离，处理器能够使用动态功率管理，而不影响 PLL、RTC 或其它 I/O 设备。

处理器的功耗主要是处理器时钟和工作电压的平方的函数。例如，降低 25%的时钟频率就能降低 25%的功耗，而降低 25%的电压可以降低 40%以上的功耗。此外，这些功率节省是加性的，如果时钟频率和电源都降低，功耗将会更显著地降低。

表 3. 运行模式功率设置

模式	PLL	PLL 旁路	内核时钟(CCLK)	系统时钟(SCLK)
全速	使能	否	使能	使能
活动	使能	是	使能	使能
休眠	使能	是或否	禁用	使能
深度休眠	禁用	-	禁用	禁用

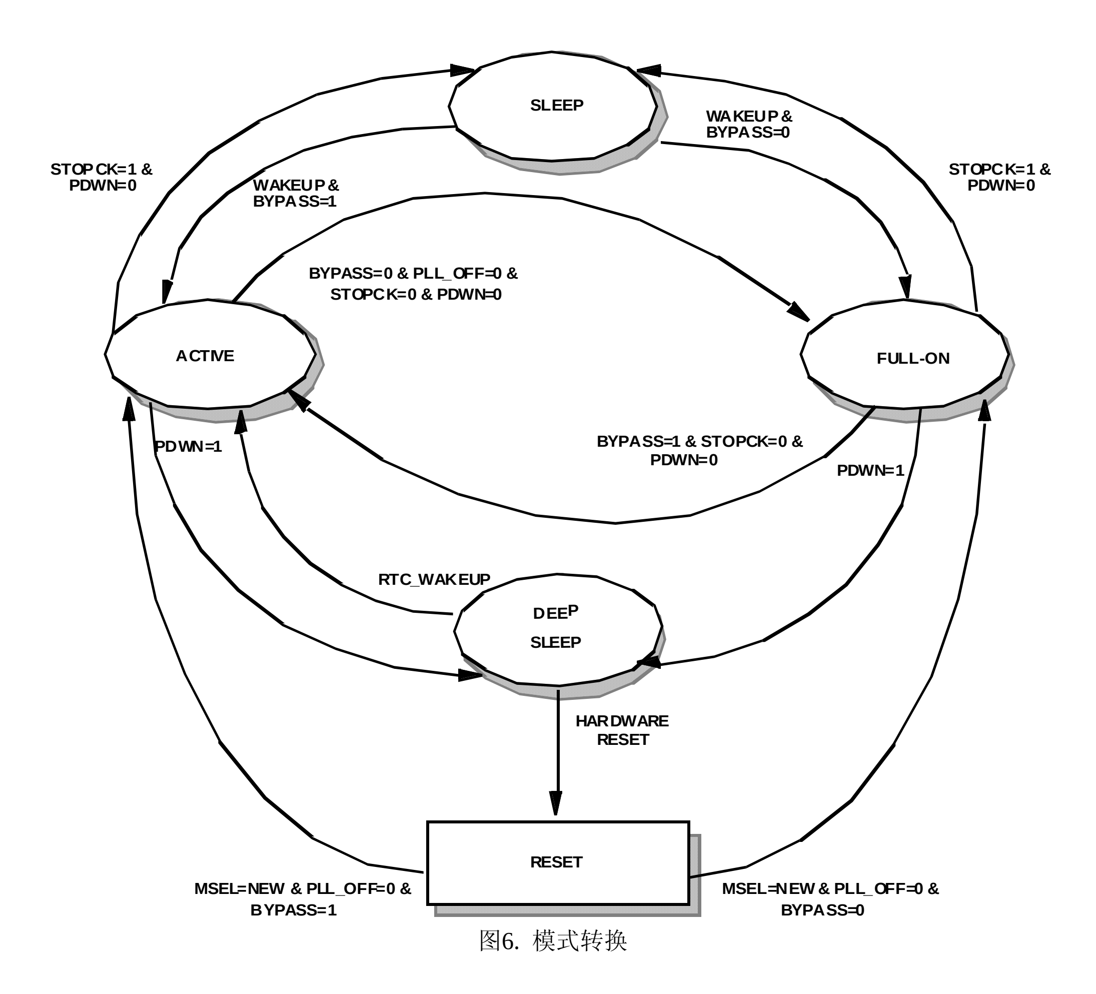


图6. 模式转换

表4. 电源范围

电源范围	**VDD范围**
所有内部逻辑（除PLL和RTC）	V_{DDINT}
模拟PLL内部逻辑	V_{DDPLL}
RTC内部逻辑和晶体I/O	V_{DDRTC}
PCI I/O	$V_{DDPCIEXT}$
所有其它I/O	V_{DDEXT}

动态功率管理允许处理器的输入电压（V_{DDINT}）和时钟频率（ f_{CLK} ）动态可控。

如上所述，功耗节省可用下面的模型：

功耗因子=

$$(f_{cclkred}/f_{cclknom})\times(V_{DDINTRED}/V_{DDINTNOM})^2$$

这里

$f_{cclknom}$ 是标称内核时钟频率（300MHZ）

$f_{cclkred}$ 是降低的内核时钟频率

$V_{DDINTNOM}$ 是标称内部供给电压（1.5V）

$V_{DDINTRED}$ 是降低的内部供给电压

举一个当频率和电压都降低时动态功率管理如何显著节省功率的例子。如果频率从标称值降到50MHZ，同时电压从标称值降到1.2V，此时，处理器的功耗只有标称频率和电压下大约10%的功耗。

外设功率控制

ADSP-21535 通过动态管理每个外设的时钟输入提供额外功率控制能力。下面列出的每个外设的时钟都可以通过适当设置外设时钟使能寄存器 PLL_IOCK 的外设控制位进行使能或禁止。外设时钟使能寄存器允许独立控制下列每个外设：

1. PCI
2. EBIU 控制器

3. 可编程标志
4. MEMDMA 控制器
5. SPORT 0
6. SPORT 1
7. SPI 0
8. SPI 1
9. UART 0
10. UART 1
11. TIMER 0，TIMER 1，TIMER 2
12. USB CLK

时钟信号

ADSP-21535 使用来自外部晶振的正弦输入，或经过缓冲整形的时钟。

如果使用缓冲整形的时钟，此外部时钟应连到 DSP 的 CLKIN 引脚。在正常运行时，CLKIN 输入不能停止、改变、或低于指定的频率。这个时钟信号应是 TTL 兼容信号。DSP 提供了用户编程的 1x 到 31x 的输入时钟倍频，即外部到内部（DSP 内核）的时钟比率。MSEL6-0、BYPASS 和 DF 引脚决定复位时 PLL 倍频因子。运行时，倍频因子可用软件控制。如图 7 所示的上拉和下拉电阻的组合设置了一个 6:1 的内核时钟比，即如果输入 25MHZ 的时钟将产生 150MHZ 的内核时钟。其它时钟倍频因子的设置，请参看 ADSP-21535 DSP Hardware Reference。

外设时钟通过 CLKOUT_SCLK0 引脚提供。

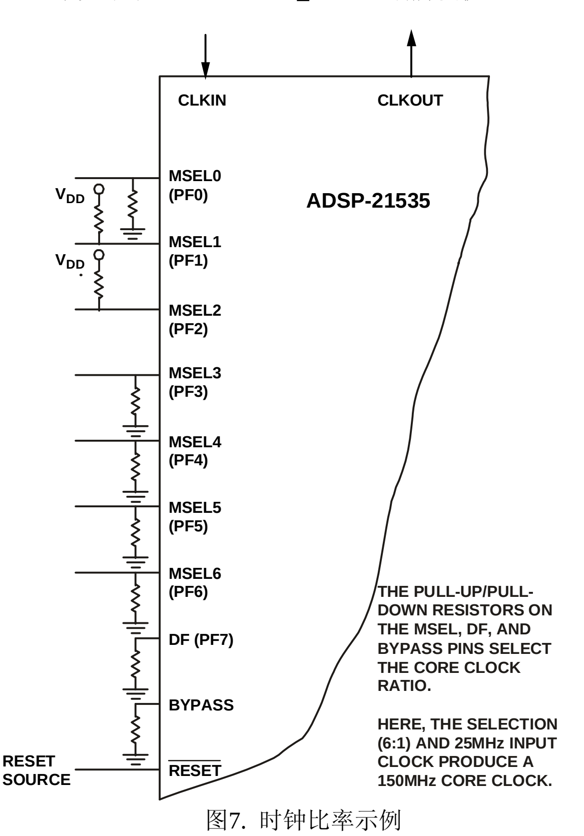


图7. 时钟比率示例

所有片上外设都以设定好的系统时钟（SCLK）速度运行。此系统时钟频率可通过SSEL编程。运行时，通过写PLL控制寄存器（PLL_CTL）中的SSEL域可以软件控制系统时钟频率。SSEL的编程值规定了内核时钟频率（CCLK）和系统时钟的分频比率。表5给出了系统时钟比率。

表5. 系统时钟比率

信号名称 SSEL-0	分频比率 CCLK/SCLK	频率比率示例 (MHz) CCLK	SCLK
00	2:1	266	133
01	2.5:1	275	110
10	3:1	300	100
11	4:1	300	75

系统时钟最高频率是 f_{SCLK}。注意分频比率必须选择，使之能将系统时钟频率限制到最大值 f_{SCLK}。复位后SSEL1-0值通过采样可编程标志输入引脚（PF9-8）来决定。SSEL的值可以通过写PLL控制寄存器（PLL_CTL）来动态改变，请参看ADSP-21535 DSP Hardware Reference。

引导模式

复位后，ADSP-21535有三种机制自动载入内部L2存储器。第四种模式从外部存储器执行，绕过引导程序。

表6. 引导模式

BMODE2–0	说明
000	从外部16位存储器执行（旁路引导ROM）
001	由8位Flash引导
010	由SPI0串行ROM引导（8位地址范围）
011	由SPI0串行ROM引导（16位地址范围）
100-111	保留

上电复位或软件复位后，处理器采样复位配置寄存器的BMODE脚，执行以下引导模式：

1. 从外部16位存储器执行—从16位紧缩地址0x2000000开始执行，此模式跳过引导ROM。

2. 从外部8位Flash引导—位于引导ROM存储空间的8位Flash引导程序通过异步存储器Bank0设置。所有的设置参数都适合最慢的设备。（3个周期的保持时间；15个周期的R/W访问时间；4个周期的建立时间。）

3. 从SPI串行EEPROM引导（8位寻址）—SPI0用PF10输出引脚选择一个SPI EPROM设备，在地址0x00开始读数据，将之存入L2存储器的起始部分。必须使用一个8位寻址的SPI兼容EPROM。

4. 由SPI串行EEPROM引导（16位寻址）—SPI0用PF10输出引脚选择一个SPI EPROM设备，从地址0x0000开始读数据，将之存入L2存储器的起始部分。必须使用一个16位寻址的SPI兼容EPROM。

无论上述何种引导模式，从存储器设备首先要读入一个四个字节的值。这个值用来指定之后要被读到 L2 存储器起始部分的字节数。一但这些字节装载完成，处理器跳到 L2 存储器空间起始位置并开始执行。

此外，软件代码可以设置复位配置寄存器，使软件复位时跳过正常的引导顺序，处理器直接从 L2 存储空间的起始部分开始执行。

为扩充上述引导模式，ADSP-21535提供了一个辅助的软件加载器以加入其它的引导功能。辅助的加载器提供的引导能力能够从16位Flash、快速Flash及可变的波特率等引导。

指令集描述

Blackfin DSP 系列汇编语言指令集使用易于编程和可读性强的代数语法，采用专门调整的灵活的高密度编码的指令，这些指令被编译后只占用非常小的存储空间。指令集还提供了体现 ADSP-21535 全部特点的多功能指令，以使在一个时钟周期内能够使用尽可能多的处理器内核资源。除许多微控制器上常见的功能外，指令集编译 C 和 C++源代码时效率非常高。此外，指令集还支持用户（算法/应用代码）和管理员（O/S 内核，设备驱动，调试器，ISRs）两种运行模式，允许对 DSP 内核资源的多级访问。

汇编语言结合了处理器独特的体系结构，具有以下优点：

1．无缝集成DSP/CPU特征，对8位和16位操作进行了优化。

2．超级流水线并发加载/存储的改进哈佛体系结构，每个周期支持两个16位MAC或4个8位ALU+两个加载/存储+两个指针更新。

3． 所有的寄存器、I/O、和存储器被映射成为统一4GB存储空间，提供了一个简化编程模式。

4． 微控制器特征，如任意位和位域操作，插入和提取；对8位、16位和32位数据类型上的整数操作；独立的用户和内核堆栈指针。

5． 代码密度增强，包括混合的16和32位指令（无模式转换和代码分离）。常用指令以16位编码。

开发工具

ADI 公司提供一套完整的 ADSP-21535 软硬件开发工具，包括仿真器和 Visual DSP++™ 开发环境。仿真器能够对 ADSP-21535 实现全仿真，还支持对其它 ADI 的 JTAG DSPs 芯片的仿真。

利用 Visual DSP++项目管理环境，程序员可以开发和调试应用程序。这个环境包括一个基于代数语法的易于使用的汇编器，一个归档器（库和库建立工具），一个链接器，一个加载器，一个精确到时钟周期、指令级的模拟器，一个 C/C++编译器和一个包括 DSP 和数学函数的 C/C++运行库。这些工具的最重要特点是 C/C++代码的有效性。编译器能有效地将 C/C++代码转换为 Blackfin DSP 的汇编代码。Blackfin DSP 体系结构的特点也提高了 C/C++代码的编译效率。

Visual DSP++调试器有许多重要的功能。绘图功能非常灵活，增强了数据的可视化能力。这种用户数据的图形化显示使程序员能快速确定一个算法的性能。随着算法复杂性的增加，这个功能对与设计者开发进度的重要性也不断提高。统计性能分析功能使程序员可以在不干扰处理器运行的情况下访问处理器。这一 Visual DSP++独特的功能使程序员能以被动方式获取重要代码的执行尺度而不中断程序的实时特性。从本质上讲，开发者可以快速而有效的找出软件的瓶颈所在。通过使用性能分析工具，程序员可以将注意力集中到影响程序性能的区域并采取相应纠正措施。

用 Visual DSP++调试器调试 C/C++和汇编程序时，程序员能够：

1. 查看混合的C/C++和汇编代码（交叉资源和目标信息）
2. 插入断点
3. 根据寄存器、存储器和堆栈设置条件断点
4. 跟踪指令执行
5. 对执行的程序进行线性或统计性能分析
6. 对存储器进行Fill、Dump和绘图操作
7. 执行源程序级调试
8. 创建用户调试窗口

Visual DSP++ IDE 能够让程序员定义和管理 DSP 软件开发环境。它的对话框和属性页可以让程序员配置和管理所有的开发工具，包括 Visual DSP++编辑器里的语法高亮颜色。这些功能允许程序员：

1. 控制开发工具如何处理输入和产生输出
2. 维护与命令行操作一一对应的操作

Visual DSP++内核（VDK）提供进度和资源管理功能，能够根据存储器地址和 DSP 程序时序约束进行裁剪，能使工程师在开发新程序时提高编程效率，不用从头写起。VDK 的特征包括线程，Critical 和 Unscheduled region，Semaphores，事件和设备标志。VDK 同时也支持优先级的、抢占的、协作和时间片的方法。此外，VDK 是可裁剪的。如果应用程序不使用某种功能，可以将支持此功能的代码从目标系统中清除。

因为 VDK 是一个库，开发者可以决定是否使用它。VDK 已被集成到了 Visual DSP++开发环境中，也可以通过标准的命令行工具使用。当使用 VDK 调试一个程序时，开发环境帮助开发者解决许多易犯的错误同时帮助管理系统资源，自动配置许多 VDK 基础目标的阶段，并可视化

系统状态。

ADI 公司的 DSP 仿真器使用 IEEE 1149.1 JTAG 测试访问端口，在仿真时监视和控制目标板上的 DSP 处理器。仿真器提供全速仿真，允许查看和修改存储器、寄存器和处理器堆栈。使用处理器 JTAG 接口可确保进行不干扰处理器运行的在电路仿真——仿真器不会影响目标系统的装载和定时。

除 ADI 公司提供软硬件开发工具外，第三方也提供了许多支持 Blankfin DSP 系列的工具。硬件工具包括 ADSP-21535 EZ-KIT LITE™ 评估/开发卡，软件工具包括 DSP 库、实时操作系统和方框图设计工具等。

设计仿真器兼容的 DSP 目标板

ADI 公司的仿真器是每个 DSP 开发者都需要的测试和调试软硬件系统的工具。ADSP-21535 提供了一个 IEEE 1149.1 JTAG 测试访问端口（TAP)。通过此 TAP，仿真器能够访问 DSP 的内部，允许开发者装载代码，设置断点，观察变量和存储器，检查寄存器。在发送数据和命令时 DSP 必须暂停，但当仿真器完成此次操作，DSP 系统就可以全速运行，不影响系统的定时时序。

要使用这些仿真器，用户目标板上必须设计有 AD 的 JTAG DSP 接口和仿真插座。

目标板仿真插座

仿真器与 AD 的 JTAG DSP 的接口是一个 14 个引脚的插座，如图 8 所示。用户必须在目标板上提供这个与仿真器通讯的接口。接口由标准的双排 0.025″ 的方型直立引脚组成， 0.1″ × 0.1″ 的间隔，0.235″ 的最小直立高度。引脚 3 这个重要的位置用来防止将仿真头插反，目标板上的这个引脚必须切断。

而且，插座周围的空隙（长、宽和高）也必须考虑。长和宽至少要分别留出 0.15″ 和 0.10″ 的空隙，高度的空隙要能够插拔仿真器的插头。

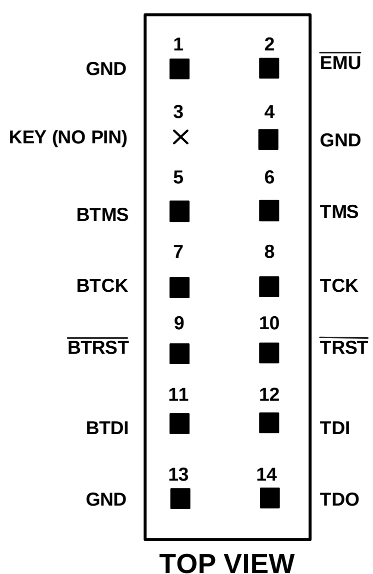


图 8. AD DSP JTAG 目标板插座

由图 8 可见，插座上有两组信号。一组是仿真用的标准 JTAG 信号 TMS，TCK，TDI，TDO，TRST 和 EMU。另一组是可选用的辅助 JTAG 信号 BTMS，BTCK，BTDI 和 BTRST，用于板级（边界扫描）的测试。

当仿真器不与目标板连接时，应如图 9 所示将跳线跨接在 BTMS，BTCK，BTRST，和 BTDI 引脚上。这将使 JTAG 信号保持正确状态从而使 DSP 自由运行。当与仿真器连接时，应移去所有的跳线。

JTAG 仿真插头

图 10 详细给出了 14 个引脚目标端的 JTAG 仿真插头的尺寸。图 11 所示的是应给目标板上插座的保留区域。保留区域可使仿真插头合适地插在目标板上。

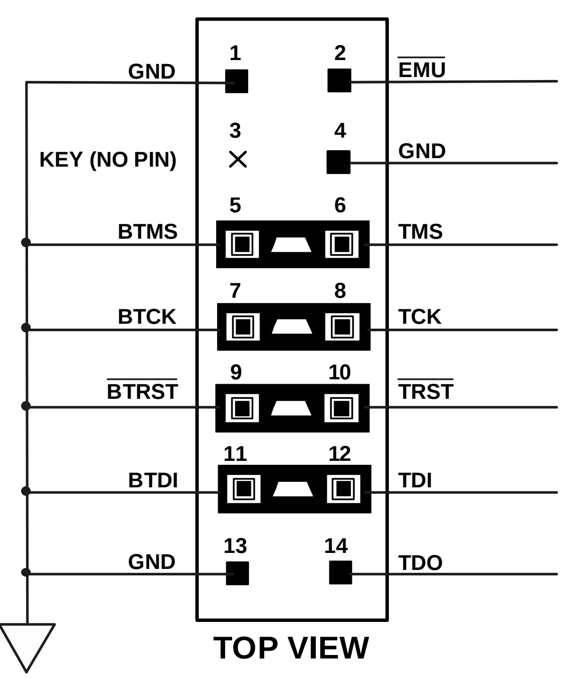


图 9. 无局部边界扫描的 JTAG 目标板插座

目标板上地保留区域不应放任何元器件（芯片、电阻、电容等)。图上尺寸以0.025"方型直立引脚的中心为参考点。

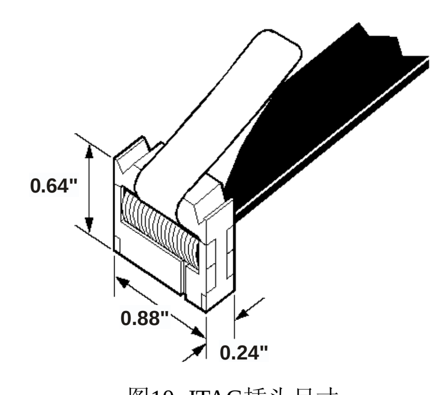


图10. JTAG插头尺寸

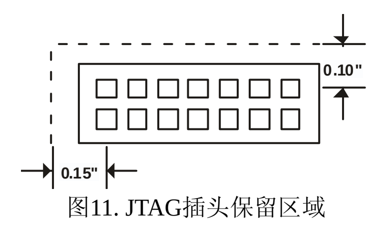


图11. JTAG插头保留区域

设计仿真电路信息

关于目标板的设计问题包括：单处理器连接，多处理器扫描链，信号缓冲，信号端接，仿真器逻辑等，详见EE-68：Analog Device JTAG Emulation Technical Refence（请到AD网站WWW.ANALOG.COM使用"EE-68"搜索）。此文档随仿真器的改进而更新。

补充信息

本数据手册为ADSP-21535的体系结构和功能提供了一个总体概览。关于Blankfin DSP系列内核体系结构和指令集的详细信息，请参考*ADSP-21535 Hardware Reference*和*Blackfin DSP Family Instruction Set Reference*。

引脚描述

ADSP-21535 的引脚定义列于表 7。下列引脚是异步的：ARDY、PF15–0、USB_CLK、NMI、TRST、RESET、PCI_CLK、XTALI、XTALO。

下列符号出现在表 7 的“类型”一栏中：I = Input、O = Output、T = Three-State、P = Power、G = Ground。

表 7. 引脚描述

引脚	类型	功能
ADDR25–2	O/T	外部地址总线
DATA31–0[1]	I/O/T	外部数据总线
$\overline{\text{ABE}}$/SDQM3–0	O/T	异步存储器字节使能，SDRAM 数据屏蔽
$\overline{\text{AMS3-0}}$[1]	O/T	异步存储器片选
ARDY[1,2]	I	异步存储器确认信号
$\overline{\text{AOE}}$	O/T	异步存储器输出使能
$\overline{\text{ARE}}$	O	异步存储器读使能
$\overline{\text{AWE}}$	O	异步存储器写使能
CLKOUT/SCLK1	O	SDRAM 时钟输出引脚，频率和定时同 SCLK0，用于减小 SCLK0 上的容性负载，连接到 SDRAM 的 CK 引脚
SCLK0	O	SDRAM 时钟输出引脚 0，系统时钟频率开关，连接到 SDRAM 的 CK 引脚
SCKE	O/T	SDRAM 时钟使能引脚。连接到 SDRAM 的 CKE 引脚
SA10	O/T	SDRAM A10 引脚，SDRAM 接口利用它保持在主机总线请求时对 SDRAM 器件的控制，连接到 SDRAM 的 A10 引脚
$\overline{\text{SRAS}}$	O/T	SDRAM 行地址选通引脚。接 SDRAM 的 RAS 引脚
$\overline{\text{SCAS}}$	O/T	SDRAM 列地址选择引脚。接 SDRAM 的 CAS 引脚
$\overline{\text{SWE}}$	O/T	SDRAM 写使能引脚。接 SDRAM 的 WE 或 W 缓冲引脚
$\overline{\text{SMS3-0}}$	O/T	配置 SDRAM 的外部存储器 bank 选择引脚，接 SDRAM 的片选引脚
TMR0[2]	I/O/T	定时器 0 引脚，PWMOUT 模式下用作输出，WIDTH_CNT 和 EXT_CLK 模式下用作输入
TMR1[2]	I/O/T	定时器 1 引脚，PWMOUT 模式下用作输出，WIDTH_CNT 和 EXT_CLK 模式下用作输入
TMR2[2]	I/O/T	定时器 2 引脚，PWMOUT 模式下用作输出，WIDTH_CNT 和 EXT_CLK 模式下用作输入
PF15/$\overline{\text{SPI1SEL7}}$[2]	I/O/T	可编程标志引脚，SPI 输出选择引脚
PF14/$\overline{\text{SPI0SEL7}}$[2]	I/O/T	可编程标志引脚，SPI 输出选择引脚
PF13/$\overline{\text{SPI1SEL6}}$[2]	I/O/T	可编程标志引脚，SPI 输出选择引脚
PF12/$\overline{\text{SPI0SEL6}}$[2]	I/O/T	可编程标志引脚，SPI 输出选择引脚
PF11/$\overline{\text{SPI1SEL5}}$[2]	I/O/T	可编程标志引脚，SPI 输出选择引脚
PF10/$\overline{\text{SPI0SEL5}}$[2]	I/O/T	可编程标志引脚，SPI 输出选择引脚（SPI 引导时使用）
PF9/$\overline{\text{SPI1SEL4}}$/SSEL1[3]	I/O	可编程标志引脚，SPI 输出选择引脚，复位时采样，以确定内核时钟与系统时钟之比
PF8/$\overline{\text{SPI0SEL4}}$/SSEL0[3]	I/O	可编程标志引脚，SPI 输出选择引脚，复位时采样，以确定内核时钟与系统时钟之比
PF7/$\overline{\text{SPI1SEL3}}$/DF[3]	I/O	可编程标志引脚，SPI 输出选择引脚，硬件复位时采样以确定配置状态，用于配置 PLL。DF=1 时将高频率输入时钟除以 2；DF=0 时，输入时钟直通到 PLL 鉴相器
PF6/$\overline{\text{SPI0SEL3}}$/MSEL6[3]	I/O	可编程标志引脚，SPI 输出选择引脚，硬件复位时采样以确定配置状态，用于配置 PLL，选择 CK 对 CLKIN 的比率
PF5/$\overline{\text{SPI1SEL2}}$/MSEL5[3]	I/O	可编程标志引脚，SPI 输出选择引脚，硬件复位时采样以确定配置状态，用于配置 PLL，选择 CK 对 CLKIN 的比率
PF4/$\overline{\text{SPI0SEL2}}$/MSEL4[3]	I/O	可编程标志引脚，SPI 输出选择引脚，硬件复位时采样以确定配置状态，用于配置 PLL，选择 CK 对 CLKIN 的比率

PF3/$\overline{\text{SPI1SEL1}}$/MSEL3 [3]	I/O	可编程标志引脚，SPI 输出选择引脚，硬件复位时采样以确定配置状态，用于配置 PLL，选择 CK 对 CLKIN 的比率
PF2/$\overline{\text{SPI0SEL1}}$/MSEL2 [3]	I/O	可编程标志引脚，SPI 输出选择引脚，硬件复位时采样以确定配置状态，用于配置 PLL，选择 CK 对 CLKIN 的比率
PF1/$\overline{\text{SPISS1}}$/MSEL1 [3]	I/O	可编程标志引脚，SPI 输出选择引脚，硬件复位时采样以确定配置状态，用于配置 PLL，选择 CK 对 CLKIN 的比率
PF0/$\overline{\text{SPISS0}}$/MSEL0 [3]	I/O	可编程标志引脚，SPI 输出选择引脚，硬件复位时采样以确定配置状态，用于配置 PLL，选择 CK 对 CLKIN 的比率
RSLCK0 [2]	I/O/T	SPORT0 串行时钟接收
RFS0 [2]	I/O/T	SPORT0 帧同步接收
DR0 [3]	I	SPORT0 串行数据接收
TSCLK0 [2]	I/O/T	SPORT0 串行时钟发送
TFS0 [2]	I/O/T	SPORT0 帧同步发送
DT0	O	SPORT0 串行数据发送
RSCLK1 [2]	I/O/T	SPORT1 串行时钟接收
RFS1 [2]	I/O/T	SPORT1 帧同步接收
DR1 [3]	I	SPORT1 串行数据接收
TSCLK1 [2]	I/O/T	SPORT1 串行时钟发送
TFS1 [2]	I/O/T	SPORT1 帧同步发送
DT1	O	SPORT1 串行数据发送
MOSI0 [2]	I/O	SPI0 主输出从输入引脚，发送来自主设备的输出数据和接收去往从设备的输入数据
MISO0 [2]	I/O	SPI0 主输入从输出引脚，发送来自从设备的输出数据和接收去往主设备的输入数据
SCK0 [4]	I/O	SPI0 时钟线，主设备输出时钟，从设备输入时钟
MOSI1 [2]	I/O	SPI1 主输出从输入引脚，发送来自主设备的输出数据和接收去往从设备的输入数据
MISO1 [2]	I/O	SPI1 主输入从输出引脚，发送来自从设备的输出数据和接收去往主设备的输入数据
SCK1 [4]	I/O	SPI1 时钟线，主设备输出时钟，从设备输入时钟
RX0 [3]	I	UART0 接收
TX0	O	UART0 发送
RX1 [3]	I	UART1 接收
TX1	O	UART1 发送
USB_CLK [4]	I	USB 时钟
XVER_DATA [4]	I	单端接收 USB 收发器到 USBD 模块的数据输出
DPLS [4]	I	差分 D+接收 USB 收发器到 USBD 模块的数据输出
DMNS [4]	I	差分 D-接收 USB 收发器到 USBD 模块的数据输出
TXDPLS	O	从 USBD 模块到 USB 收发器发送 D+
TXDMNS	O	从 USBD 模块到 USB 收发器发送 D-
$\overline{\text{TXEN}}$	O	从 USBD 模块到 USB 收发器发送使能
SUSPEND	O	从 USBD 模块到 USB 收发器暂停模式使能输出，该信号也能被路由至系统内部，以支持低功耗运行
NMI [4]	I	不可屏蔽中断
TCK [2]	I	JTAG 时钟
TDO	O/T	JTAG 串行数据输出
TDI [2]	I	JTAG 串行数据输入
TMS [2]	I	测试模式选择

$\overline{\text{TRST}}$ [4]	I	JTAG 复位
$\overline{\text{RESET}}$	I	低电平至少 10 CLKIN 周期，硬件复位开始；上电复位最小脉冲宽度 40μS
CLKIN1	I	时钟输入
BYPASS	I	专用模式引脚，可永久与 V_{DD} 或 V_{SS} 连接，旁路片内 PLL
DEEPSLEEP	O	表明 Blackfin DSP 内核处于深度休眠运行模式
BMODE2-0	I	专用模式引脚，可永久与 V_{DD} 或 V_{SS} 连接，配置硬件或软件复位后的引导模式
PCI_AD31-0 [2]	I/O/T	PCI 地址和数据总线
$\overline{\text{PCI_CBE3 - 0}}$ [2]	I/O/T	PCI 字节使能
$\overline{\text{PCI_FRAME}}$ [2]	I/O/T	PCI 帧信号，表明 PCI 交易的开始和结束，被 PCI 启动方使用
$\overline{\text{PCI_IRDY}}$ [2]	I/O/T	PCI 启动方准备好信号
$\overline{\text{PCI_TRDY}}$ [2]	I/O/T	PCI 目标准备好信号
PCI_DEVSEL [2]	I/O/T	PCI 设备选择信号，PCI 交易目标用该信号的有效声明交易
$\overline{\text{PCI_STOP}}$ [2]	I/O/T	PCI 停止信号
$\overline{\text{PCI_PERR}}$ [2]	I/O/T	PCI 奇偶校验错误信号
PCI_PAR [2]	I/O/T	PCI 奇偶校验信号
$\overline{\text{PCI_REQ}}$ [2]	O	PCI 请求信号，用于请求使用 PCI 总线
$\overline{\text{PCI_SERR}}$ [2]	I/O/T	PCI 系统错误信号，在系统板上要求上拉
$\overline{\text{PCI_RST}}$ [2]	I/O/T	PCI 复位信号
$\overline{\text{PCI_GNT}}$ [2]	I	PCI 允许信号，用于允许访问 PCI 总线
$\overline{\text{PCI_IDSEL}}$ [2]	I	PCI 初始化设备选择信号，PCI 配置传输期间选择不同的目标设备
$\overline{\text{PCI_LOCK}}$ [2]	I	PCI 锁定信号，用于锁定一个目标或整个 PCI 总线，供要锁定的主机用
PCI_CLK [4]	I	PCI 时钟
$\overline{\text{PCI_INTA}}$ [2]	I/O/T	PCI 总线的 PCI 中断 A 线；当 ADSP-21535 作为一个设备时，发出此中断信号给系统处理器；当 ADSP-21535 作为系统处理器时监视此引脚
$\overline{\text{PCI_INTB}}$ [2]	I	PCI 中断 B 线，当 ADSP-21535 作为系统处理器时监视此引脚
$\overline{\text{PCI_INTC}}$ [2]	I	PCI 中断 C 线，当 ADSP-21535 作为系统处理器时监视此引脚
$\overline{\text{PCI_INTD}}$ [2]	I	PCI 中断 D 线，当 ADSP-21535 作为系统处理器时监视此引脚
XTALI	I	实时时钟晶振输入。
XTALO	O	实时时钟晶振输出。
$\overline{\text{EMU}}$	O	仿真器确认，漏极开路，必须连接到 ADSP-21535 仿真器与目标板相接的仿真插座
$\overline{V_{DDPLL}}$	P	PLL 电源（1.5V 标称）
$\overline{V_{DDRTC}}$	P	实时时钟电源（3.3V 标称）
$\overline{V_{DDEXT}}$	P	I/O（PCI 除外）电源（3.3V 标称）
$\overline{V_{DDPCIEXT}}$	P	PCI I/O 电源（3.3V 标称）
$\overline{V_{DDINT}}$	P	内部电源（1.5V 标称）
GND	G	地

[1] 引脚有逻辑电平保持电路，防止输入内部浮置。

[2] 如果不用，引脚上拉。

[3] 如果不用，引脚上拉或下拉。

[4] 如果不用，引脚下拉。

ADSP-21535 —— 技术规格

推荐操作条件

参数[1]	K 级参数	最小	标称	最大	单位
V_{DDINT}	内部（内核）供电电压	0.86	1.5	1.575	V
V_{DDEXT}	外部（I/O）供电电压	2.5	3.3	3.45	V
V_{DDPLL}	PLL 电源电压	1.425	1.5	1.575	V
V_{DDRTC}	实时时钟电源电压	2.60	3.3	3.45	V
$V_{DDPCIEXT}$	PCI I/O 电源电压	3.15	3.3	3.45	V
V_{IH}	高电平输入电压[2]，@V_{DDEXT}=max	2.0		V_{DDEXT}+0.5	V
V_{IL}	低电平输入电压[2]，@V_{DDEXT}=min	-0.3		0.6	V
V_{IHPCI}	高电平输入电压[3]，@V_{DDEXT}=max	0.5$V_{DDPCIEXT}$		$V_{DDPCIEXT}$+0.5	V
V_{ILPCI}	低电平输入电压[3]，@V_{DDINT}=min	-0.5		0.3 $V_{DDPCIEXT}$	V
T_{CASE}	外壳工作温度	0		85	ºC

[1] 规格有改动不另行通知。

[2] 适用于输入和双向引脚，PCI 除外。

[3] 适用于PCI输入和双向引脚：PCI_AD31–0，PCI_CBE3-0，PCI_FRAME，PCI_IRDY，PCI_TRDY，PCI_DEVSEL，PCI_STOP，PCI_PERR，PCI_PAR，PCI_SERR，PCI_RST，PCI_GNT，PCI_IDSEL，PCI_LOCK，PCI_CLK，PCI_INTA，PCI_INTB，PCI_INTC，PCI_INTD。

电气特性

参数[1]		测试条件	最小	最大	单位
V_{OH}	高电平输出电压[2]	@ V_{DDEXT} =min, I_{OH} = –0.5 mA	2.4		V
V_{OL}	低电平输出电压[2]	@ V_{DDEXT} =min, I_{OL} = 2.0 mA		0.4	V
V_{OHPCI}	PCI高电平输出电压[3]	@ V_{DDEXT} =min, I_{OH} = –0.5 mA	0.9 $V_{DDPCIEXT}$		V
V_{OLPCI}	PCI低电平输出电压[3]	@ V_{DDEXT} =min, I_{OL} = 1.5mA		0.1 $V_{DDPCIEXT}$	V
I_{IH}	高电平输入电流[4]	@ V_{DDEXT} =max, V_{IN} = V_{DD} max		TBD	μA
I_{IL}	低电平输入电流[4]	@ V_{DDEXT} =max, V_{IN} = 0 V		TBD	μA
I_{OZH}	三态漏电流[5]	@ V_{DDEXT} = max, V_{IN} = V_{DD} max		TBD	μA
I_{OZL}	三态漏电流[5]	@ V_{DDEXT} = max, V_{IN} = 0 V		TBD	μA
C_{IN}	输入电容[6,7]	f = 1 MHz, T_{CASE} = 25°C, V_{IN} = 2.5 V		TBD	pF

[1] 规格有改动不另行通知

[2] 适用于输入和双向引脚，PCI 除外。

[3] 适用于PCI输入和双向引脚：PCI_AD31–0，PCI_CBE3-0，PCI_FRAME，PCI_IRDY，PCI_TRDY，PCI_DEVSEL，PCI_STOP，PCI_PERR，PCI_PAR，PCI_REQ，PCI_SERR，PCI_RST，PCI_INTA。

[4] 用于输入引脚。

[5] 用于三态引脚。

[6] 用于所有信号引脚。

[7] 保证但未测试。

绝对最大额定值

内部（内核）供电电压（V_{DDINT}）[1] ……-0.3V 到 1.8V
外部（I/O）供电电压（V_{DDEXT}）……-0.3V 到 4.0V
输入电压…………………………-0.5V 到 V_{DDEXT}+0.5V
输出电压漂移………………………-0.5V 到 V_{DDEXT}+0.5V
负载电容………………………………………………200pF
内核时钟频率…………………………………………300MHz
外设时钟频率(SCLK)………………………………133MHz
储存温度范围……………………………-65°C 到 150°C
管脚温度（5 秒）………………………………………185°C

[1] 过载大于上述值时芯片会受到永久性损坏。以上只是过载额定值，芯片在超出上述条件下的工作情况没有给出。在最大额定值条件下持续工作可能影响芯片的可靠性。

ESD 灵敏度

注意

ESD（静电放电）灵敏器件。静电容易在人体或测试设备上积累高达 4000V 且放电不宜察觉。虽然 ADSP-21535 有 ESD 保护电路，但当芯片受到高能静电放电时可能会受到永久性破坏。因此，采用适当的 ESD 保护措施，以防止芯片性能降低和功能损失。



时序规格

表 8 和表 9 描述了 ADSP-21535 时钟的时序规格。注意选择 MSEL 和 SSEL 的比率使之不要超出最大内核时钟和系统时钟工作频率，参照绝对最大额定值表。表 9 描述了锁相环工作条件。

表 8. 内核与系统时钟需要

参数	说明	最大	最小	单位
$t_{CCLK1.5}$	内核周期（V_{DDINT}=1.5V-5%）	3.3	TBD	ns
$t_{CCLK1.4}$	内核周期（V_{DDINT}=1.4V-5%）	TBD	TBD	ns
$t_{CCLK1.3}$	内核周期（V_{DDINT}=1.3V-5%）	TBD	TBD	ns
$t_{CCLK1.2}$	内核周期（V_{DDINT}=1.2V-5%）	TBD	TBD	ns
$t_{CCLK1.1}$	内核周期（V_{DDINT} =1.1V-5%）	TBD	TBD	ns
$t_{CCLK1.0}$	内核周期（V_{DDINT} =1.0V-5%）	TBD	TBD	ns
$t_{CCLK0.9}$	内核周期（V_{DDINT} =0.9V-5%）	TBD	TBD	ns
f_{CCLKNN}	内核时钟频率：t_{CCLKNN}		1/t_{CCLKNN}	Hz
t_{SCLK}	系统时钟周期	最大（7.5 或 t_{CCLKNN}×2）		ns
f_{SCLK}	系统时钟频率		1/t_{SCLK}	Hz

表 9. 锁相环工作条件

参数	最小	一般	最大	单位
工作电压	1.425	1.5	1.575	V
抖动,上升沿到上升沿(每个输出)			120	ps
抖动,下降沿到上升沿(每个输出)			60	ps
偏斜,上升沿到上升沿(任意两个输出)			120	ps
压控振荡器(VCO)频率	40		400	MHz
V_{DDPLL} 引入的抖动			1	ps/mV

时钟和复位时序

表 10 和图 12 描述了时钟和复位操作。在上述绝对最大额定值的工作条件下，CLKIN 和时钟倍频器的选择必须使内核/外设时钟不超过 300/133MHz。

表 10. 时钟和复位时序

参数	说明	最小	最大	单位
时序要求				
t_{CKIN}	CLKIN 周期	30.0	100.0	ns
t_{CKINL}	CLKIN 脉冲低电平 [1]	10.0		ns
t_{CKINH}	CLKIN 脉冲高电平 [1]	10.0		ns
t_{WRST}	$\overline{RESET}$ 有效低电平脉宽 [2]	11×t_{CKIN}		ns
t_{PFD}	从 $\overline{RESET}$ 有效到 PFx I/O 终止的延迟 [3]		TBD	ns
t_{MSD}	从 $\overline{RESET}$ 有效到 MSELx 和 DF 有效的延迟 [4]		TBD	ns
t_{MSS}	$\overline{RESET}$ 失效前 MSELx/DF/BYPASS 建立稳定 [5]	TBD		ns
t_{MSH}	$\overline{RESET}$ 失效后 MSELx/DF/BYPASS 保持稳定	TBD		ns
开关特性				
t_{SCLKD}	从 CLKIN 到 CLKOUT 的延迟	TBD	TBD	ns
t_{SCLK}	CLKOUT 周期 [6]	7.5		ns

[1] 用于旁路模式和非旁路模式.

[2] 用于上电顺序已完成情况。在上电时且当复位有效，处理器内部锁相环需要不超过 2000 个 CLKIN 周期，假定电源稳定供给和 CLKIN（不包括外部时钟振荡器的启动时间）。

[3] 从这一点，MSELx 和 DF 值开始稳定到一个有效状态。

[4] MSELx 和 DF 值可以从这一点变化，但值必须有效。

[5] MSELx 和 DF 值必须从这时保持，直到持续时间结束。

[6] 下图按 t_{CKIN} 和 t_{SCLK}×2 比率显示，但比率可以有很多可编程的选择。参照 *ADSP-21535 Hardware Reference*。

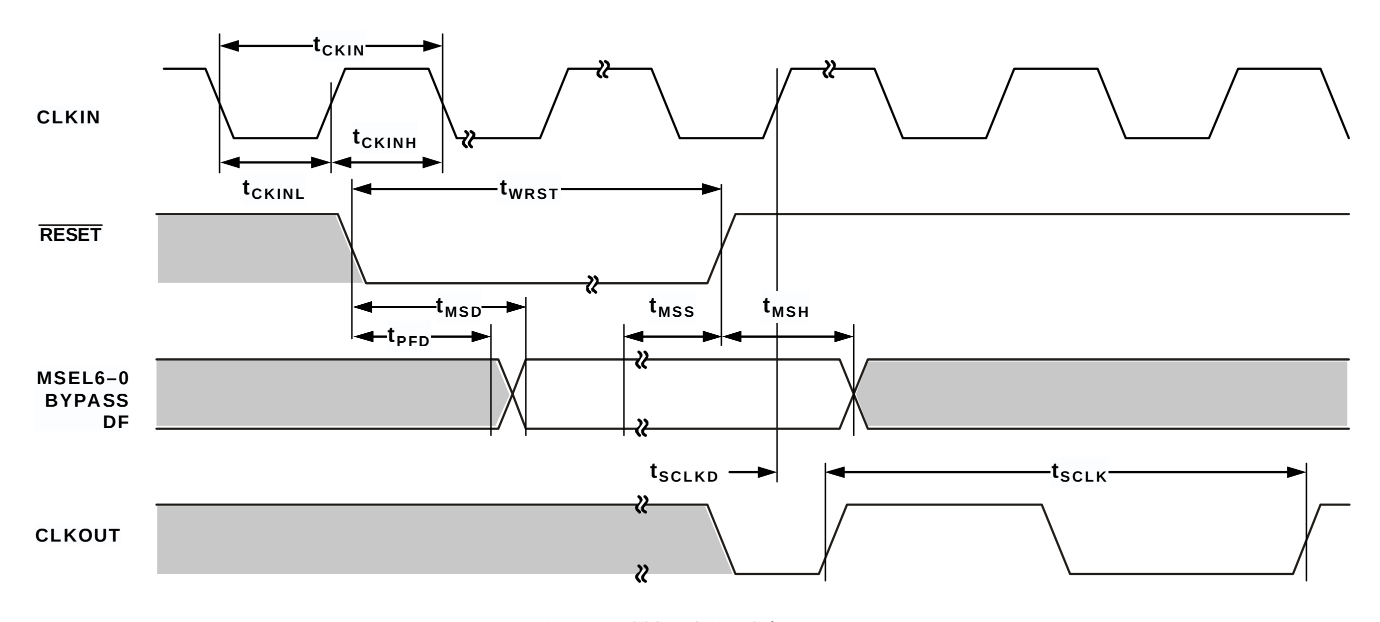


图 12. 时钟和复位时序

可编程标志周期时序

表 11 和图 13 描述了可编程标志操作。

表 11. 可编程标志周期时序

参数	说明	Min	Max	单位
开关特性				
t_{DFO}	相对于 SCLK 的标志输出延迟		6	ns
t_{HFO}	SCLK 高后的标志输出维持	TBD	TBD	ns
时序要求				
t_{HFI}	标志输出保持为异步	3		ns

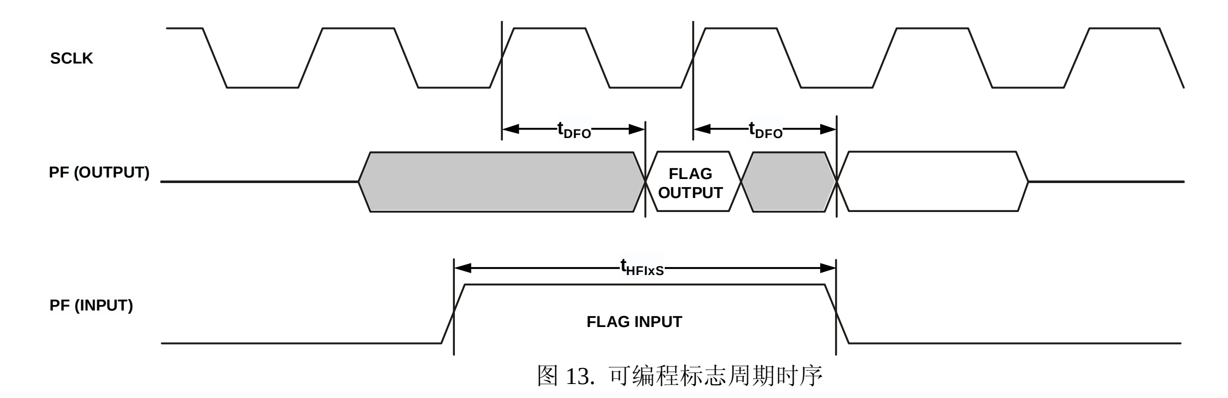


图 13. 可编程标志周期时序

定时器 PWM_OUT 周期时序

表 12 和图 14 描述了定时器计数满操作。

输入信号在宽度捕获模式（width capture mode）下是异步的，且有一个 TBD MHz 的绝对最大输入频率。

表 12. 定时器 PWM_OUT 周期时序

参数	说明	最小	最大	单位
开关特性				
t_{HTO}	定时器脉宽输出 [1]	7.5	（$2^{32}-1$）个周期	ns

[1]t_{HTO} 的最小时间是一个周期，t_{HTO} 的最大时间为（$2^{32}-1$）个周期。

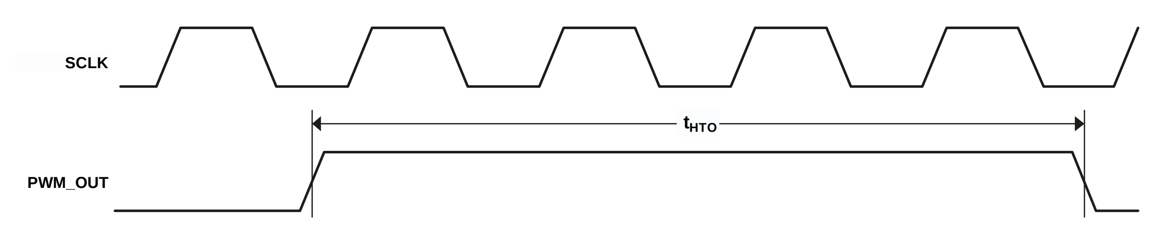


图 14. 定时器 PWM_OUT 周期时序

异步存储器写周期时序

表 13. 异步存储器写周期时序

参数	说明	最小	最大	单位
时序要求				
t_{SARDY}	CLKOUT 前 ARDY 建立	5.5		ns
t_{HARDY}	CLKOUT 后 ARDY 保持	0.0		ns
t_{DDAT}	CLKOUT 后 DATA31-0 关闭		6.0	ns
t_{ENDAT}	CLKOUT 后 DATA31-0 打开	1.0		ns
开关特性				
t_{DO}	CLKOUT 后输出延迟 [1]		6.0	ns
t_{HO}	CLKOUT 后输出保持 [1]	0.8		ns

[1] 输出引脚包括 $\overline{\text{AMS3-0}}$，$\overline{\text{ABE3-0}}$，ADDR25-2，DATA31-0，$\overline{\text{AOE}}$，$\overline{\text{AWE}}$。

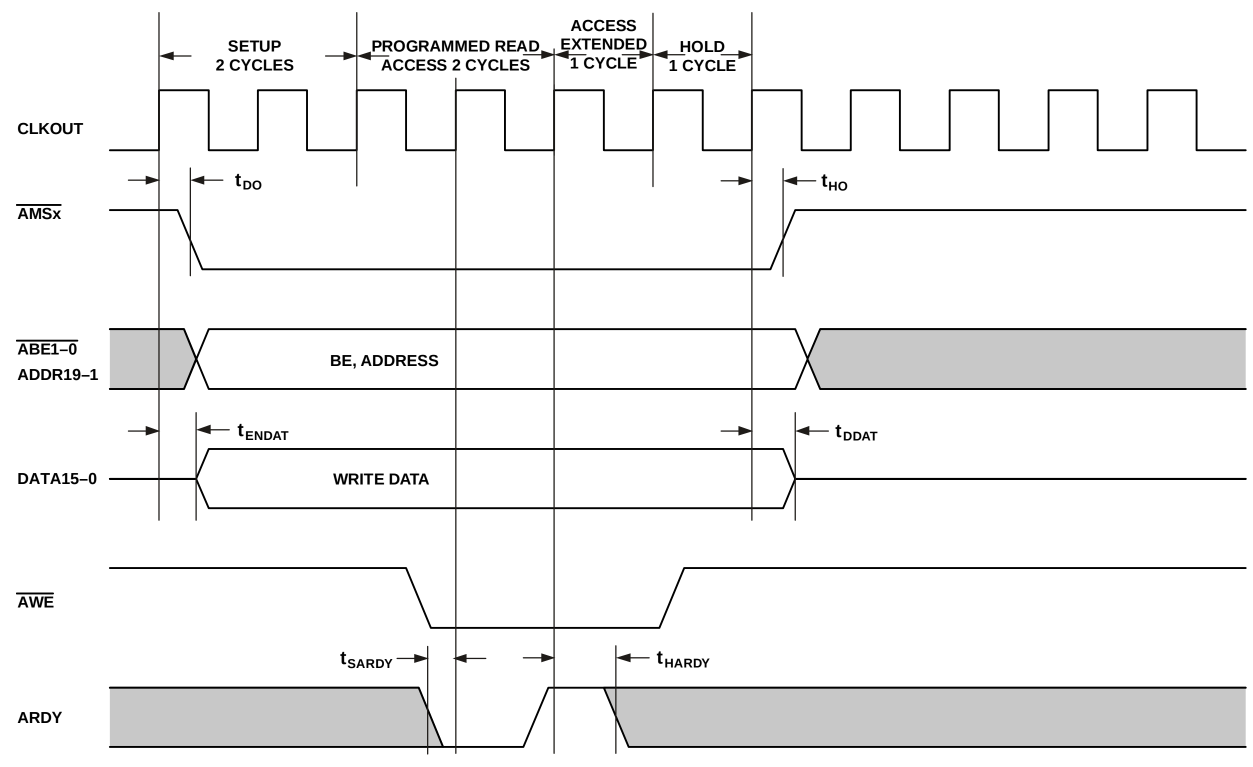


图 15. 异步存储器写周期时序

异步存储器读周期时序

表 14. 异步存储读周期时序

参数	说明	最小	最大	单位
时序要求				
t_{SDAT}	CLKOUT 前 DATA31-0 建立	2.1		ns
t_{HDAT}	CLKOUT 后 DATA31-0 保持	0.8		ns
t_{SARDY}	CLKOUT 前 ARDY 建立	5.5		ns
t_{HARDY}	CLKOUT 后 ARDY 保持	0.0		ns
开关特性				
t_{DO}	CLKOUT 后输出延迟 [1]		6.0	ns
t_{HO}	CLKOUT 后输出保持 [1]	0.8		ns

[1] 输出引脚包括 $\overline{AMS3\text{-}0}$， $\overline{ABE3\text{-}0}$，ADDR25-2， $\overline{AOE}$， $\overline{ARE}$ 。

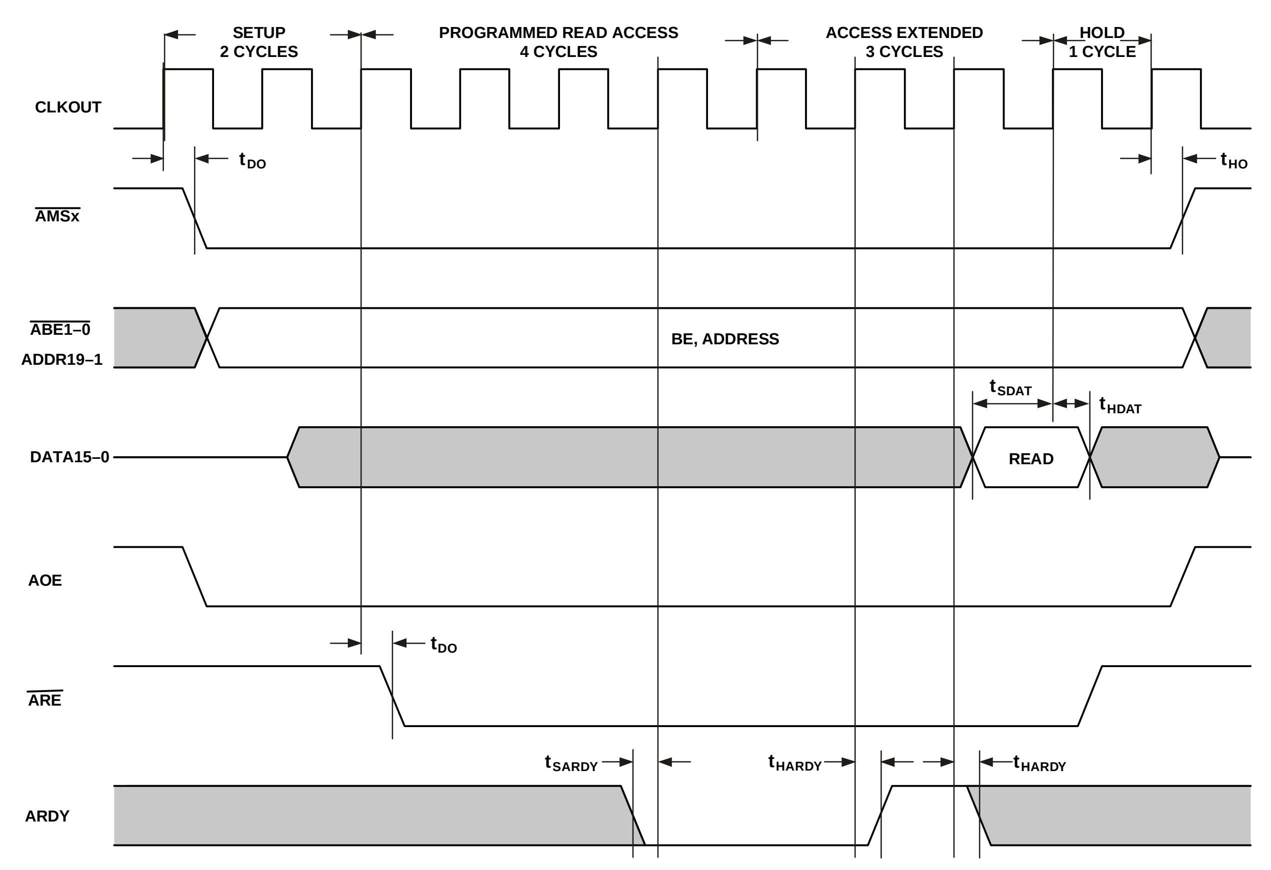


图 16. 异步存储器读周期时序

SDRAM 接口时序

表 15. SDRAM 接口时序

参数	说明	最小	最大	单位
时序要求				
t_{SSDAT}	CLKOUT 前 DATA 建立	2.1		ns
t_{HSDAT}	CLKOUT 后 DATA 保持	0.8		ns
开关特性				
t_{SCLK}	CLKOUT 周期	7.5		ns
t_{SCLKH}	CLKOUT 高电平宽度	TBD		ns
t_{SCLKL}	CLKOUT 低电平宽度	TBD		ns
t_{DCAD}	CLKOUT 后命令、地址和数据延迟 [1]		6.0	ns
t_{HCAD}	CLKOUT 后命令、地址和数据保持 [1]	0.8		ns
t_{DSDAT}	CLKOUT 后 DATA31-0 关闭		6.0	ns
t_{ENSDAT}	CLKOUT 后 DATA31-0 使能	1.0		ns

[1] 命令引脚包括：$\overline{\text{SRAS}}$，$\overline{\text{SCAS}}$，$\overline{\text{SWE}}$，SDQM，$\overline{\text{SMS}}$，SA10，SCKE。

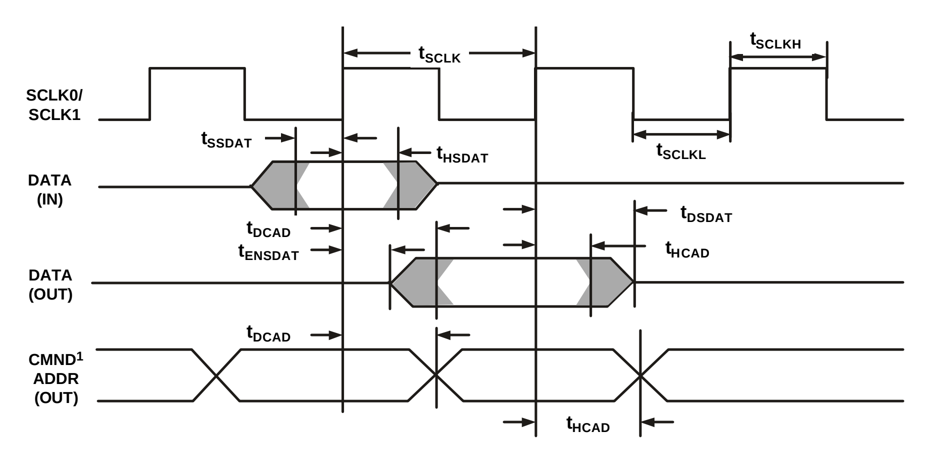


图 17. SDRAM 接口时序

串行口

表 16. 串行口—外部时钟

参数	说明	最小	最大	单位
时序要求				
t_{SFSE}	TCLK/RCLK 前 TFS/RFS 建立 [1]	3.0		ns
t_{HFSE}	TCLK/RCLK 后 TFS/RFS 保持 [1]	3.0		ns
t_{SDRE}	RCLK 前接收数据建立 [1]	3.0		ns
t_{HDRE}	RCLK 后接收数据保持 [1]	3.0		ns
t_{SCLKW}	TCLK/RCLK 宽度	4.5		ns
t_{SCLK}	TCLK/RCLK 周期	15.0		ns

[1] 以采样边沿为参考

表 17. 串行口—内部时钟

参数	说明	最小	最大	单位
时序要求				
t_{SFSI}	TCLK/RCLK 前 TFS/RFS 建立 [1]	7.0		ns
t_{HFSI}	TCLK/RCLK 后 TFS/RFS 保持 [1]	2.0		ns
t_{SDRI}	RCLK 前接受数据建立 [1]	7.0		ns
t_{HDRI}	RCLK 后接受数据保持 [1]	4.0		ns

[1] 以采样边沿为参考

表 18. 串行口—内部或外部时钟

参数	说明	最小	最大	单位
开关特性				
t_{DFSE}	RCLK 后 RFS 延迟（内部产生的 RFS）[1]		10.0	ns
t_{HOFSE}	RCLK 后 RFS 保持（内部产生的 RFS)[1]	6.0		ns

[1] 以驱动边沿为参考

表 19. 串行口—外部时钟

参数	说明	最小	最大	单位
开关特性				
t_{DFSE}	TCLK 后 TFS 延迟（内部产生的 TFS）[1]		10.0	ns
t_{HOFSE}	TCLK 后 TFS 保持（内部产生的 TFS）[1]	6.0		ns
t_{DDTE}	TCLK 后发送数据延迟		10.0	ns
t_{HDTE}	TCLK 后发送数据保持 [1]	6.0		ns

[1] 以驱动边沿为参考

表 20. 串行口—内部时钟

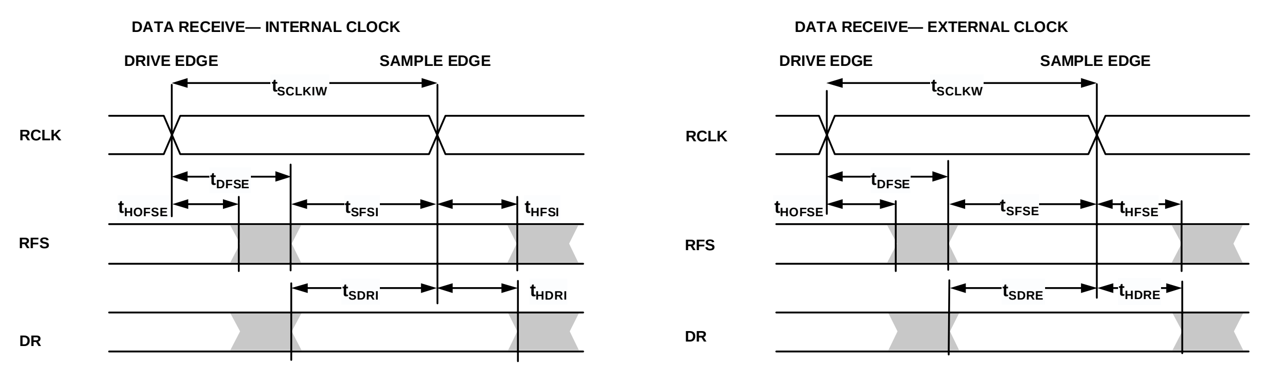
参数	说明	最小	最大	单位
开关特性				
t_{DFS1}	TCLK 后 TFS 延迟（内部产生的 TFS）[1]		4.0	ns
t_{HOFS1}	TCLK 后 TFS 保持（内部产生的 TFS）[1]	0.0		ns
t_{DDT1}	TCLK 后发送数据延迟		4.0	ns
t_{HDT1}	TCLK 后发送数据保持 [1]	0.0		ns
t_{SCLKIW}	TCLK/RCLK 宽度	4.5		ns

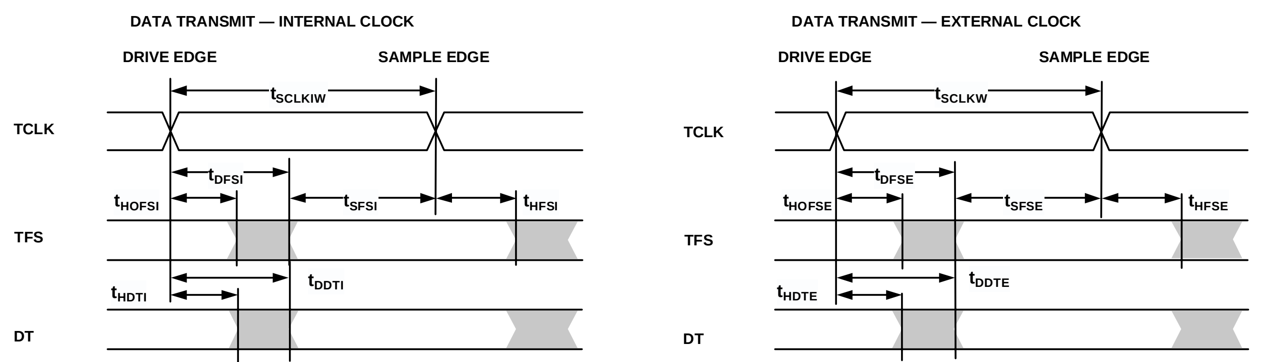
[1] 以驱动边沿为参考

表 **21.** 串行口—使能和三态

参数	说明	最小	最大	单位
开关特性				
t_{DTENE}	外部 TCLK 的数据使能延迟 [1]	5.0		ns
t_{DDTTE}	外部 TCLK 的数据关闭延迟 [1]		12.0	ns
t_{DTENI}	内部 TCLK 的数据使能延迟	2.0		ns
t_{DDTTI}	内部 TCLK 的数据关闭延迟 [1]		5.0	ns

[1] 以驱动边沿为参考





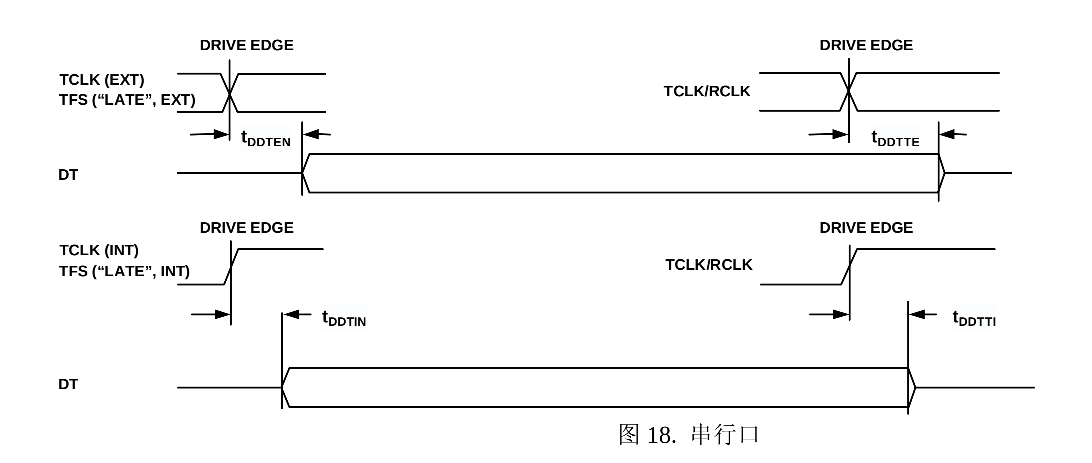


图 18. 串行口

串行外设接口（SPI）口—主时序

表 22 和图 19 描述 SPI 端口主操作。

表 22. 串行外设接口（SPI）口—主时序

参数	说明	最小	最大	单位
时序要求				
t_{SSPID}	数据输入有效到 SCLK 边沿（数据输入建立）	1.6		ns
t_{HSPID}	SCLK 采样边沿到数据输入无效	1.6		ns
开关特性				
t_{SDSCIM}	$\overline{\text{SPIxSEL}}$ 低到第一个 SCLK 边沿（x=0 或 1）	$2t_{SCLK}$		ns
t_{SPICHM}	串行时钟高电平时间	$2t_{SCLK}$		ns
t_{SPICLM}	串行时钟低电平时间	$2t_{SCLK}$		ns
t_{SPICLK}	串行时钟周期	$4t_{SCLK}$		ns
t_{HDSM}	最后 SCLK 边沿到 $\overline{\text{SPIxSEL}}$ 高（x=0 或 1）	$2t_{SCLK}$		ns
t_{SPITDM}	序列传输延迟	$2t_{SCLK}$		ns
t_{DDSPID}	SCLK 边沿到数据输出有效（数据输出延迟）	0	6	ns
t_{HDSPID}	SCLK 边沿到数据输出无效（数据输出保持）	0	5	ns

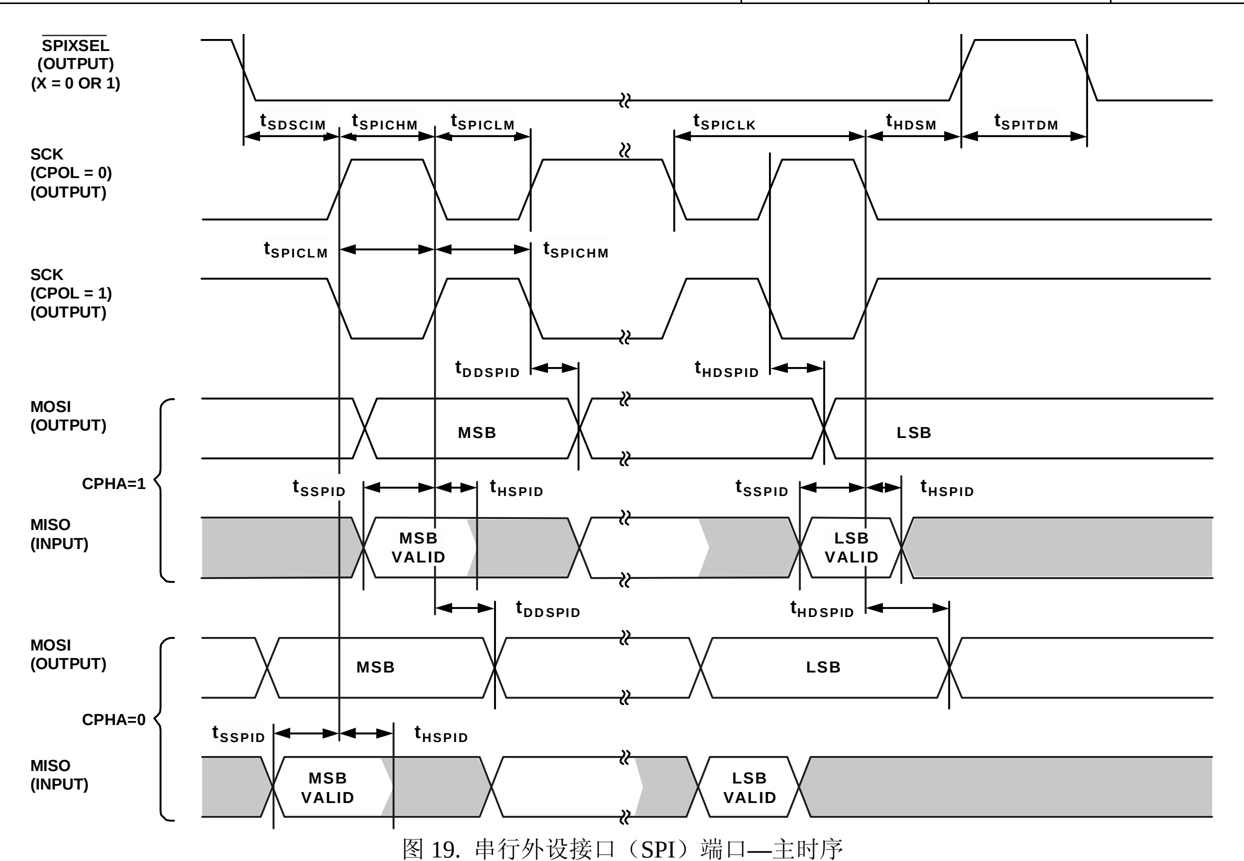


图 19. 串行外设接口（SPI）端口—主时序

串行外设接口（SPI）端口—从时序

表 23 和图 20 描述了 SPI 端口从操作。

表 23. 串行外设接口（SPI）端口—从时序

参数	说明	最小	最大	单位
时序要求				
t_{SPICHS}	串行时钟高电平时间	$2t_{SCLK}$		ns
t_{SPICLS}	串行时钟低电平时间	$2t_{SCLK}$		ns
t_{SPICLK}	串行时钟周期	$4t_{SCLK}$		ns
t_{HDS}	最后一个 SPICLK 边沿到 $\overline{SPISS}$ 无效	$2t_{SCLK}$		ns
t_{SPITDS}	序列传输延迟	$2t_{SCLK}$		ns
t_{SDSCI}	$\overline{SPISS}$ 有效到第一个 SCLK 边沿	$2t_{SCLK}$		ns
t_{SSPID}	数据输入有效到 SCLK 边沿（数据输入建立）	1.6		ns
t_{HSPID}	SCLK 采样边沿到数据输入无效	1.6		ns
开关特性				
t_{DSOE}	$\overline{SPISS}$ 有效到数据输出启动	0	6	ns
t_{DSDHI}	$\overline{SPISS}$ 无效到数据高阻抗	0	6	ns
t_{DDSPID}	SCLK 边沿到数据输出有效（数据输出延时）	0	5	ns
t_{HDSPID}	SCLK 边沿到数据输出无效（数据输出保持）	0	5	ns

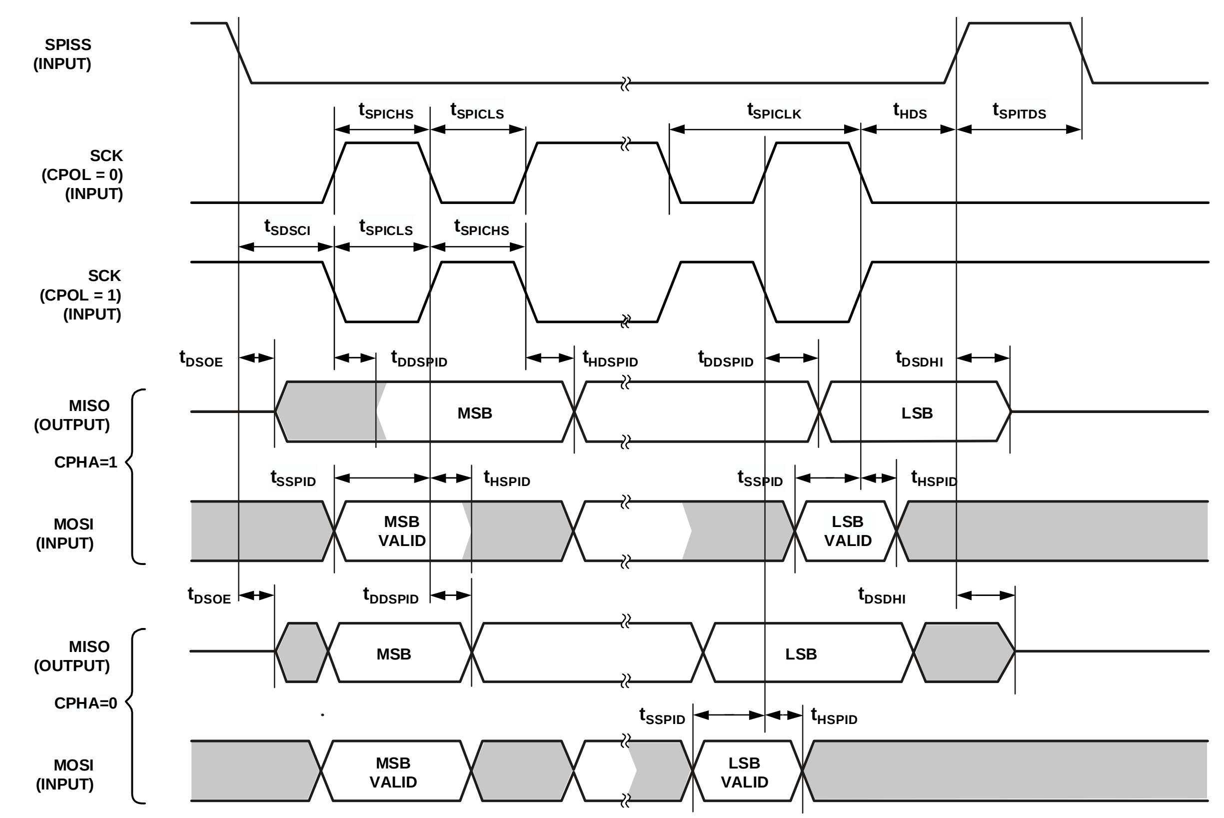


图 20. 串行外设接口（SPI）端口—从时序

通用串行收发器（UART）端口—接收和发送时序

图 21 描述了 UART 端口接收和发送操作，最大波特率是 SCLK/16。如图 21 所示，在产生内部 UART 中断和外部数据操作之间有一些延迟。这些延迟同 UART 数据传输率相比可以忽略。

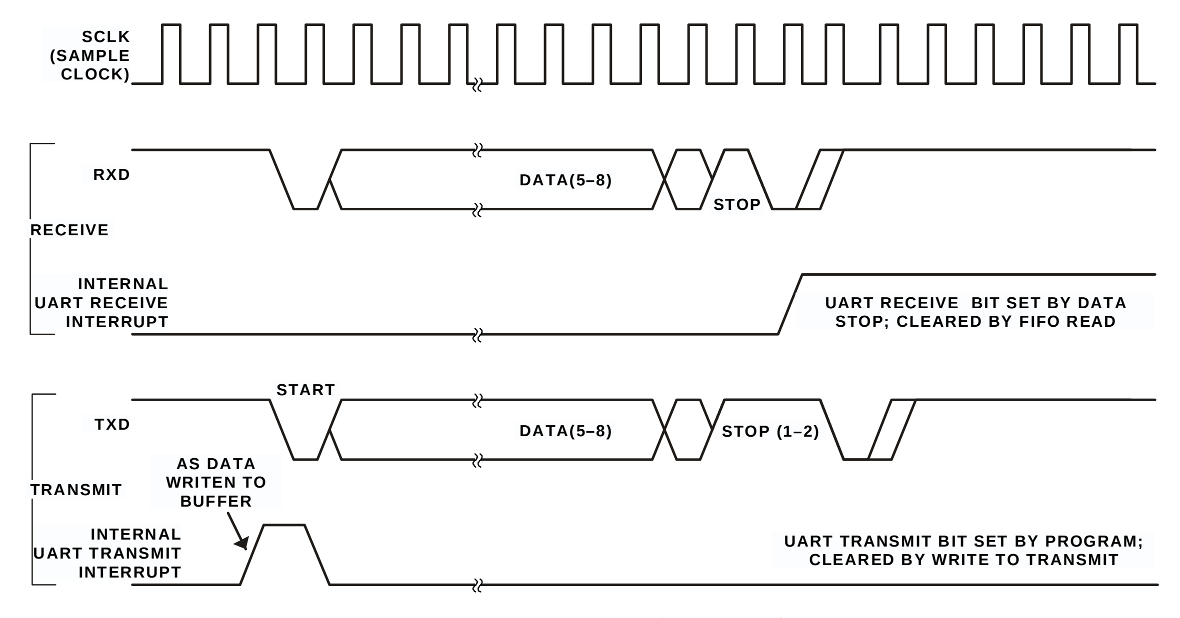


图 21. UART 端口—接收和发送时序

JTAG 测试和仿真端口时序

表 24 和图 22 描述 JTAG 端口的操作。

表 24. JTAG 端口时序

参数	说明	最小	最大	单位
时序参数				
t_{TCK}	TCK 周期	20		ns
t_{STAP}	TCK 高电平前 TDI 和 TMS 建立		4	ns
t_{HTAP}	TCK 高电平后 TDI 和 TMS 保持		4	ns
t_{SSYS}	TCK 低电平前系统输入建立 [1]		4	ns
t_{HSYS}	TCK 低电平后系统输入保持 [1]		5	ns
t_{TRSTW}	$\overline{TRST}$ 脉冲宽度 [2]	4		ns
开关特性				
t_{DTDO}	TCK 低电平后 TDO 延迟		4	ns
t_{DSYS}	TCK 低电平后系统输出延迟 [3]	0	5	ns

[1]系统输入=DATA31-0, ADDR25-2, ARDY, TMR2-0, PF15-0, RSCLK0, RFS0, DR0, TSCLK0, TFS0, RSCLK1, RFS1, DR1, TSCLK1, TFS1,MOSI0, MISO0, SCK0, MOSI1, MISO1, SCK1, RX0, RX1, TSB_CLK, XVER_DATA, DPLS, DMNS, NMI, $\overline{RESET}$, BYPASS, BMODE2-0,PCI_AD31-0, $\overline{PCI_CBE3-0}$, $\overline{PCI_FRAME}$, $\overline{PCI_IRDY}$, $\overline{PCI_TRDY}$, PCI_DEVSEL, $\overline{PCI_STOP}$, $\overline{PCI_PERR}$, PCI_PAR, $\overline{PCI_SERR}$, $\overline{PCI_RST}$, $\overline{PCI_GNT}$, $\overline{PCI_IDSEL}$, $\overline{PCI_LOCK}$, PCI_CLK, $\overline{PCI_INTA}$, $\overline{PCI_INTB}$, $\overline{PCI_INTC}$, $\overline{PCI_INTD}$。

[2]50 MHz max。

[3]System Outputs=DATA31-0, ADDR25-2, $\overline{ABE}$/SDQM3-0, $\overline{AOE}$, $\overline{ARE}$, $\overline{AWE}$, CLKOUT/SCLK1, SCLK0, SCKE, SA10, $\overline{SRAS}$, $\overline{SCAS}$, $\overline{SWE}$, $\overline{SMS3-0}$,TMR2-0, PF15-0, RSCLK0, RFS0, TSCLK0, TFS0, DT0, RSCLK1, RFS1, TSCLK1, TFS1, DT1, MOSI0, MISO0, SCK0, MOSI1, MISO1, SCK1,TX0, TX1, TXDPLS, TXDMNS, $\overline{TXEN}$, SUSPEND, DEEPSLEEP, PCI_AD31-0, $\overline{PCI_CBE3-0}$, $\overline{PCI_FRAME}$, $\overline{PCI_IRDY}$, $\overline{PCI_TRDY}$,PCI_DEVSEL, $\overline{PCI_STOP}$, $\overline{PCI_PERR}$, PCI_PAR, $\overline{PCI_REQ}$, $\overline{PCI_SERR}$, $\overline{PCI_RST}$, $\overline{PCI_INTA}$, $\overline{EMU}$。

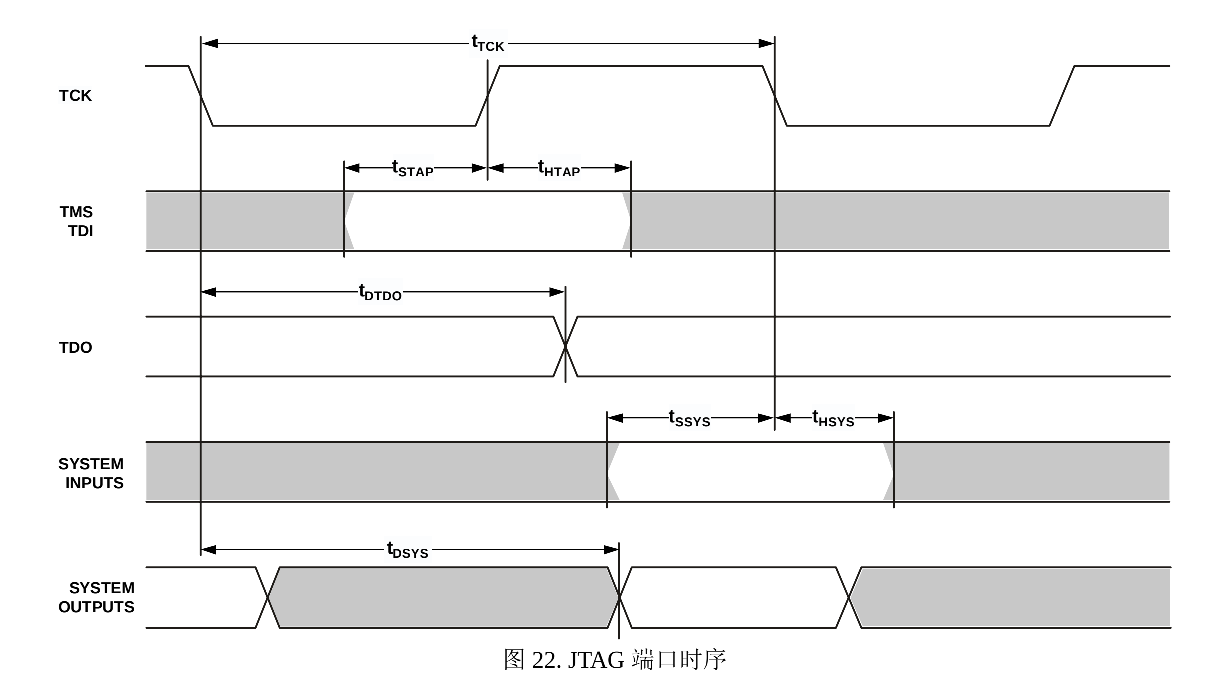


图 22. JTAG 端口时序

功耗

总功耗有两部分，一部分是内部电路（P_{INT}）的功耗，另一部分是内部输出驱动（P_{EXT}）的开关功耗。表 25 给出了内部电路的功耗。内部功耗取决于内部指令执行序列和

相关的数据操作。表 26 列出了取得表 25 给出值的条件。

表 25. 内部功耗

参数	**测试条件**	**典型值(V_{DDINT}=1.5V)[1]**	**典型值(V_{DDINT}=1.0V)[1]**	**单位**
I_{DDHIGH}	$t_{CCLKMIN}$,25ºC	TBD	TBD	mA
I_{DDTYP}	$t_{CCLKMIN}$,25ºC	TBD	TBD	mA
I_{DDLOW}	$t_{CCLKMIN}$,25ºC	TBD	TBD	mA
I_{DDSYS}	$t_{CCLKMIN}$,25ºC	TBD	TBD	mA
I_{DDEFR}	$t_{CCLKMIN}$,25ºC	TBD	TBD	mA
$I_{DDACTIVE}$	25ºC	TBD	TBD	mA
$I_{DDSLEEP}$	25ºC	TBD	TBD	mA
$I_{DDDEEPSLEEP}$	25ºC	TBD	TBD	mA

[1] 典型 I_{DD} 数据指定为标称的 V_{DDINT} 和典型处理参数。最大 I_{DD} 在典型值的 TBD%范围内。

表 26. 内部功耗条件

参数	**模式**	**PLL**	**CCLK**	**SCLK**	**单位**
I_{DDHIGH}[1]	全速	使能	使能	使能	TBD
I_{DDTYP}[1]	全速	使能	使能	使能	TBD
I_{DDLOW}[1]	全速	使能	使能	使能	TBD
I_{DDSYS}[2]	全速	使能	使能	使能	TBD
I_{DDEFR}[3]	全速	使能	使能	使能	算法有关
$I_{DDACTIVE}$	活动	使能/旁路	使能	使能	TBD
$I_{DDSLEEP}$	休眠	使能	关闭	使能	TBD
$I_{DDDEEPSLEEP}$	深度休眠	关闭	关闭	关闭	TBD

[1] TBD 指令混合。

[2] 每周期 TBD 指令混合和系统 DMA。

[3] 执行增强全速率(EFR)GSM 算法，从 L1/L2 存储器和 Cache 取指令和数据。

总功耗的外部元件部分由输出引脚的开关引起，其大小取决于

1. 每周期开关的输出引脚个数（O）
2. 最大开关频率（f）
3. 负载电容（C）
4. 电压摆动（V_{DDEXT}）

外部元件功耗用下式计算

$$P_{EXT} = O \times C \times V_{DD} \times f$$

频率 f 包括驱动负载变为高电平之后又回到低电平。例如：DATA31-0 引脚处于 SDRAM 突发模式下能以 1/(2 $\times t_{SCLK}$)的最大速率驱动高电平和低电平。

这种情况下的典型功耗应当用加上一个典型内部功耗的公式计算。

$$P_{TOTAL} = P_{EXT} + (I_{DD} \times V_{DDINT})$$

注意，引起最坏 P_{EXT} 的条件和引起最坏 P_{INT} 的的条件不同。最大 P_{INT} 不能发生在所有的输出引脚由全部 1 变为全部 0 时。同时注意，对于一个应用实例，100%，甚至 50%的输出同时开关是不常见的。

环境条件

ADSP-21535 提供 260 引脚的 PBGA 封装。

ADSP-21535 规定了外壳温度（T_{CASE}）。为保证 T_{CASE} 不超出范围，可以使用气流源。T_{CASE} 用下式计算

$$T_{CASE} = T_{AMB} + (PD \times \theta_{CA})$$

T_{CASE}=外壳温度（在封装顶面测量）

PD=功耗 W（此值取决于具体应用）

表 27 是气流每分钟线性英尺的 3 个例子。

表 27. 气流

线性 Ft/Min	0	200	400
θ_{CA}(ºC/W)	TBD	TBD	TBD

ADSP-21535 260 –Lead PBGA 的引脚

表 28 按信号名称列出了 PBGA 的引脚，表 29 给出了 PBGA 引脚的按序号排列。

表 28. 260 引脚 PBGA 排列表（按字母顺序排列）

信号	引脚 #
$\overline{ABE0}$/SDQM0	E02
$\overline{ABE1}$/SDQM1	B01
$\overline{ABE2}$/SDQM2	G03
$\overline{ABE3}$/SDQM3	H07
ADDR2	A06
ADDR3	B06
ADDR4	D06
ADDR5	C06
ADDR6	A05
ADDR7	B05
ADDR8	A04
ADDR9	C05
ADDR10	D05
ADDR11	B04
ADDR12	A01
ADDR13	C04
ADDR14	D04
ADDR15	A03
ADDR16	B03
ADDR17	A02
ADDR18	C03
ADDR19	D03
ADDR20	B02
ADDR21	C02
ADDR22	E03
ADDR23	C01
ADDR24	F03
ADDR25	D02
$\overline{AMS0}$	F02
$\overline{AMS1}$	D01
$\overline{AMS2}$	H03
$\overline{AMS3}$	G02
$\overline{AOE}$	E01
ARDY	R01
$\overline{ARE}$	F01
$\overline{AWE}$	G01
BMODE0	B14
BMODE1	A14
BMODE2	B13
BYPASS	C12
CLKIN1	D09
CLKOUT/SCLK1	H01
	N02
DATA0	N02
DATA1	M03
DATA2	T01
DATA3	P02
DATA4	N03
DATA5	R02
DATA6	P03
DATA7	U01
DATA8	U02
DATA9	T02
DATA10	V02
DATA11	V03
DATA12	R04
DATA13	U03
DATA14	T03
DATA15	T04
DATA16	U04
DATA17	V04
DATA18	V05
DATA19	R05
DATA20	T05
DATA21	U05
DATA22	V06
DATA23	R06
DATA24	U06
DATA25	T06
DATA26	V07
DATA27	V08
DATA28	U07
DATA29	R07
DATA30	T07
DATA31	V09
DMNS	D08
DPLS	C09
DR0	V14
DR1	U15
DT0	R14
DT1	V17
$\overline{EMU}$	A13
GND	C13
GND	H02
GND	H08
GND	H10
GND	H11
GND	J07
GND	J08
GND	J09
GND	J10
GND	J11
GND	J12
GND	K02
GND	K07
GND	K08
GND	K09
GND	K10
GND	K11
GND	K12
GND	L07
GND	L08
GND	L09

GND	L10
GND	L11
GND	M07
GND	M09
GND	M10
MISO0	T16
MISO1	U18
MOSI0	U16
MOSI1	T17
N/C	A18
N/C	R03
N/C	V01
N/C	V18
NMI	B11
PCI_AD0	E17
PCI_AD1	E18
PCI_AD2	G16
PCI_AD3	F17
PCI_AD4	F18
PCI_AD5	G18
PCI_AD6	G17
PCI_AD7	H18
PCI_AD8	J18
PCI_AD9	H17
PCI_AD10	K18
PCI_AD11	H16
PCI_AD12	L18
PCI_AD13	J17
PCI_AD14	M18
PCI_AD15	K17
PCI_AD16	J16
PCI_AD17	K16
PCI_AD18	N18
PCI_AD19	P18
PCI_AD20	L17
PCI_AD21	L16
PCI_AD22	R18
PCI_AD23	T18
PCI_AD24	M17
PCI_AD25	M16
PCI_AD26	N17
PCI_AD27	P17
PCI_AD28	P15
PCI_AD29	N16
PCI_AD30	R17
PCI_AD31	P16
PCI_CBE0	F16
PCI_CBE1	F15
PCI_CBE2	E16
PCI_CBE3	D17
PCI_CLK	D14
PCI_DEVSEL	C16
PCI_FRAME	C17
PCI_GNT	C18
PCI_IDSEL	B18
PCI_INTA	C14
PCI_INTB	B15
PCI_INTC	A15
PCI_INTD	D13
PCI_IRDY	E15
PCI_LOCK	A16
PCI_PAR	C15
PCI_PERR	D15
PCI_REQ	D16
PCI_RST	D18
PCI_SERR	B16
PCI_STOP	A17
PCI_TRDY	B17
PF0 /SPISS0 /MSEL0	U08
PF1 /SPISS1 /MSEL1	R08
PF2 /SPI0SEL1 MSEL2	T08
PF3 /SPI1SEL1 /MSEL3	V10
PF4 /SPI0SEL2 /MSEL4	U09
PF5 /SPI1SEL2 /MSEL5	R09
PF6 /SPI0SEL3 /MSEL6	T09
PF7 /SPI1SEL3 /DF	R11
PF8 /SPI0SEL4 /SSEL0	T11
PF9 /SPI1SEL4 /SSEL1	U11
PF10 /SPI0SEL5	V12
PF11 /SPI1SEL5	T12
PF12 /SPI0SEL6	R12
PF13 /SPI1SEL6	U12
PF14 /SPI0SEL7	V13
PF15 /SPI1SEL7	T13

$\overline{\text{RESET}}$	B09
RFS0	U13
RFS1	V16
RSCLK0	R13
RSCLK1	U14
RX0	A07
RX1	B08
SA10	M01
$\overline{\text{SCAS}}$	L03
SCK0	U17
SCK1	R16
SCKE	L01
SCLK0	K01
SLEEP	D12
$\overline{\text{SMS0}}$	M02
$\overline{\text{SMS1}}$	P01
$\overline{\text{SMS2}}$	N01
$\overline{\text{SMS3}}$	K03
$\overline{\text{SRAS}}$	L02
SUSPEND	A11
$\overline{\text{SWE}}$	J03
TCK	D10
TDI	C11
TDO	D11
TFS0	T14
TFS1	R15
TMR0	B07
TMR1	C07
TMR2	D07
TMS	A12
$\overline{\text{TRST}}$	B12
TSCLK0	V15
TSCLK1	T15
TX0	A08
TX1	C08
TXDMNS	G10
TXDPLS	B10
$\overline{\text{TXEN}}$	C10
USB_CLK	G07
V_{DDEXT}	E04
V_{DDEXT}	G04
V_{DDEXT}	G08
V_{DDEXT}	J01
V_{DDEXT}	J02
V_{DDEXT}	J04
V_{DDEXT}	K04
V_{DDEXT}	L04
V_{DDEXT}	M04
V_{DDEXT}	P04
V_{DDINT}	F04
V_{DDINT}	G11
V_{DDINT}	G12
V_{DDINT}	G15
V_{DDINT}	H04
V_{DDINT}	H09
V_{DDINT}	H12
V_{DDINT}	L12
V_{DDINT}	M08

V_{DDINT}	M11
V_{DDINT}	M12
V_{DDINT}	N04
V_{DDINT}	N15
$V_{DDPCIEXT}$	H15
$V_{DDPCIEXT}$	J15
$V_{DDPCIEXT}$	K15
$V_{DDPCIEXT}$	L15
$V_{DDPCIEXT}$	M15
V_{DDPLL}	G09
V_{DDRTC}	U10
V_{SSPLL}	A10
V_{SSRTC}	V11
XTALI	R10
XTALO	T10
XVER_DATA	A09

表 29. 260 –Lead PBGA 排列表（按引脚序号排列）

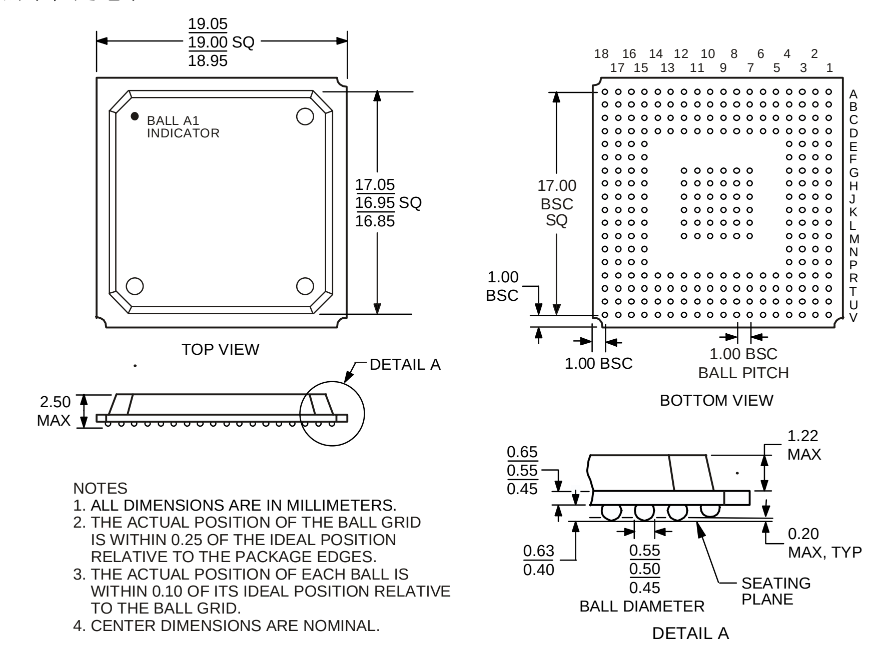
引脚#	信号
A01	ADDR12
A02	ADDR17
A03	ADDR15
A04	ADDR8
A05	ADDR6
A06	ADDR2
A07	RX0
A08	TX0
A09	XVER_DATA
A10	V_{SSPLL}
A11	SUSPEND
A12	TMS
A13	EMU
A14	BMODE1
A15	PCI_INTC
A16	PCI_LOCK
A17	PCI_STOP
A18	N/C
B01	ABE1/SDQM1
B02	ADDR20
B03	ADDR16
B04	ADDR11
B05	ADDR7
B06	ADDR3
B07	TM R0
B08	RX1
B09	RESET
B10	TXDPLS
B11	NMI
B12	TRST
B13	BMODE2
B14	BMODE0
B15	PCI_INTB
B16	PCI_SERR
B17	PCI_TRDY
B18	PCI_IDSEL
C01	ADDR23
C02	ADDR21
C03	ADDR18
C04	ADDR13
C05	ADDR9
C06	ADDR5
C07	TMR1
C08	TX1
C09	DPLS
C10	*TXEN*
C11	TDI
C12	BYPASS
C13	GND
C14	PCI_INTA
C15	PCI_PAR
C16	PCI_DEVSE
C17	PCI_FRAME
C18	PCI_GNT
D01	AMS1
D02	ADDR25
D03	ADDR19
D04	ADDR14
D05	ADDR10
D06	ADDR4
D07	TMR2
D08	DMNS
D09	CLKIN1
D10	TCK
D11	TDO
D12	SLEEP
D13	PCI_INTD
D14	PCI_CLK
D15	PCI_PERR
D16	PCI_REQ
D17	PCI_CBE3
D18	PCI_RST
E01	AOE
E02	ABED/SDQM0
E03	ADDR22
E04	V_{DDEXT}
E15	PCI_IRDY
E16	PCI_CBE2
E17	PCI_AD0
E18	PCI_AD1
F01	ARE
F02	AMS0
F03	ADDR24
F04	V_{DDINT}
F15	PCI_CBE1
F16	PCI_CBE0
F17	PCI_AD3
F18	PCI_AD4
G01	AWE
G02	AMS3
G03	ABE2/SDQM2
G04	V_{DDEXT}
G07	USB_CLK
G08	V_{DDEXT}
G09	V_{DDPLL}
G10	TXDMNS
G11	V_{DDINT}
G12	V_{DDINT}
G15	V_{DDINT}
G16	PCI_AD2
G17	PCI_AD6
G18	PCI_AD5
H01	CLKOUT/SCLK1
H02	GND
H03	AMS2
H04	V_{DDINT}
H07	ABE3/SDQM3
H08	GND
H09	VDDINT
H10	GND
H11	GND
H12	V_{DDINT}
H15	$V_{DDPCIEXT}$
H16	PCI_AD11
H17	PCI_AD9
H18	PCI_AD7
J01	V_{DDEXT}
J02	V_{DDEXT}
J03	SWE
J04	V_{DDEXT}

J07	GND
J08	GND
J09	GND
J10	GND
J11	GND
J12	GND
J15	$V_{DDPCIEXT}$
J16	PCI_AD16
J17	PCI_AD13
J18	PCI_AD8
K01	SCLK0
K02	GND
K03	$\overline{SMS3}$
K04	V_{DDEXT}
K07	GND
K08	GND
K09	GND
K10	GND
K11	GND
K12	GND
K15	$V_{DDPCIEXT}$
K16	PCI_AD17
K17	PCI_AD15
K18	PCI_AD10
L01	SCKE
L02	$\overline{SRAS}$
L03	$\overline{SCAS}$
L04	V_{DDEXT}
L07	GND
L08	GND
L09	GND
L10	GND
L11	GND
L12	V_{DDINT}
L15	$V_{DDPCIEXT}$
L16	PCI_AD21
L17	PCI_AD20
L18	PCI_AD12
M01	SA10
M02	$\overline{SMS0}$
M03	DATA1
M04	V_{DDEXT}
M07	GND
M08	V_{DDINT}
M09	GND
M10	GND
M11	V_{DDINT}
M12	V_{DDINT}
M15	$V_{DDPCIEXT}$
M16	PCI_AD25
M17	PCI_AD24
M18	PCI_AD14
N01	$\overline{SMS2}$
N02	DATA0
N03	DATA4
N04	V_{DDINT}
N15	V_{DDINT}
N16	PCI_AD29
N17	PCI_AD26
N18	PCI_AD18
P01	$\overline{SMS1}$
P02	DATA3
P03	DATA6
P04	V_{DDEXT}
P15	PCI_AD28

P16	PCI_AD31
P17	PCI_AD27
P18	PCI_AD19
R01	ARDY
R02	DATA5
R03	N/C
R04	DATA12
R05	DATA19
R06	DATA23
R07	DATA29
R08	PF1 /$\overline{SPISS1}$ /MSEL1
R09	PF5 /$\overline{SPI1SEL2}$ /MSEL5
R10	XTALI
R11	PF7 /$\overline{SPI1SEL3}$ /DF
R12	PF12 /$\overline{SPI0SEL6}$
R13	RSCLK0
R14	DT0
R15	TFS1
R16	SCK1
R17	PCI_AD30
R18	PCI_AD22
T01	DATA2
T02	DATA9
T03	DATA14
T04	DATA15
T05	DATA20
T06	DATA25
T07	DATA30
T08	PF2 /$\overline{SPI0SEL1}$ /MSEL2
T09	PF6 /$\overline{SPI0SEL3}$ /MSEL6
T10	XTALO
T11	PF8 /$\overline{SPI0SEL4}$ /SSEL0
T12	PF11 /$\overline{SPI1SEL5}$
T13	PF15 /$\overline{SPI1SEL7}$
T14	TFS0
T15	TSCLK1
T16	MISO0
T17	MOSI1
T18	PCI_AD23
U01	DATA7
U02	DATA8
U03	DATA13
U04	DATA16
U05	DATA21
U06	DATA24
U07	DATA28
U08	PF0 /$\overline{SPISS0}$

	/MSEL0
U09	PF4
	/$\overline{SPI0SEL2}$
	/MSEL4
U10	V_{DDRTC}
U11	PF9
	/$\overline{SPI1SEL4}$
	/SSEL1
U12	PF13
	/$\overline{SPI1SEL6}$
U13	RFS0
U14	RSCLK1
U15	DR1
U16	MOSI0
U17	SCK0
U18	MISO1
V01	N/C
V02	DATA10
V03	DATA11
V04	DATA17
V05	DATA18
V06	DATA22
V07	DATA26
V08	DATA27
V09	DATA31
V10	PF3
	/$\overline{SPI1SEL1}$
	/MSEL3
V11	V_{SSRTC}
V12	PF10
	/$\overline{SPI0SEL5}$
V13	PF14
	/$\overline{SPI0SEL7}$
V14	DR0
V15	TSCLK0
V16	RFS1
V17	DT1
V18	N/C

外型尺寸：图 23 的单位是毫米



注：

1. 单位：毫米
2. 球型栅格的实际位置在理想位置的 0.25mm 间距之内，相对于封装边缘而言。
3. 每个焊球的实际位置在理想位置的 0.10mm 间距之内，相对于封装边缘而言。
4. 中心尺寸是标称的。

图 23. 260-Lead 公制塑封球型栅格阵列(PBGA)（B-260）

订货指导

型号	外壳温度范围	指令速度	工作电压
ADSP-21535PKB-300	0ºC ~85ºC	300MHz	内部 0.9V~1.5V I/O 3.3V
ADSP-21535PBB-200	-40ºC ~+105ºC	200MHz	内部 0.9V~1.5V I/O 3.3V